UNIVERSITAT DE BARCELONA

Facultat de Matemàtiques i Informàtica

GRAU DE MATEMÀTIQUES

Treball final de grau

Big Data Marketing: Transformación de datos y análisis de series temporales

Autor: Jesús Llenas Puigdemont

Director:	Dr. Josep Vives
Realizado en:	Departament de Matemàtiques i Informàtica
Barcelona,	20 de junio de 2020

Abstract

Currently, data analysis is entirely incorporated in marketing, and Big Data processes are increasingly standard in these sectors. The main purpose of this work is to apply Big Data techniques and time series analysis to obtain tangible results for a marketing project. The methodology used follows three steps: first, massive data mining from Google as well as internal data; secondly, the transformation of this data and the creation of multiple data bases; and thirdly, an exhaustive analysis of all the data obtained. In terms of results, search patterns in Google have been identified and classified in accordance with the searcher's intentions. This, together with data mining, has resulted in the collection of multiple variables which are crucial pointers for future marketing campaigns. An example of this would be the percentage of people who enroll after having requested information (lead), or which Google searches lead to higher enrolment numbers. Time series have also been generated from the variables and the correlation among them has been studied. Very interesting correlations have been found such as a 0,88 in the count of users requesting information and the count of users who click-through to a website after having carried out a transactional search (transactional click-throughs). Finally, an analysis of the time transactional click-through series and a 30 week prediction based on auto-regression models for mobile media, have been carried out.

Resumen

Actualmente, el análisis de datos está totalmente integrado en marketing y cada vez las prácticas de Big Data son más habituales en estos sectores. El principal objetivo de este trabajo es utilizar técnicas de Big Data y de análisis de series temporales para obtener resultados tangibles en un proyecto real de marketing. La metodología empleada sigue tres pasos: primero, una extracción masiva de datos tanto de Google como internos de la empresa; segundo, la transformación de estos datos y la creación de diferentes bases de datos y; tercero, el análisis exhaustivo de todos los datos obtenidos. Como resultados, se han identificado patrones de búsqueda en Google y se han clasificado según la intencionalidad del buscador. Esto, junto con la extracción de datos, ha permitido la obtención de múltiples variables que serán indicadores cruciales en futuras campañas de marketing. Un buen ejemplo sería el porcentaje de gente que se matricula después de pedir información (lead) o qué búsquedas de Google acaban con más matrículas. También se han generado series temporales a partir de las variables y se ha estudiado las correlaciones entre ellas. Como resultado de este estudio se han establecido interesantes correlaciones, entre ellas cabe destacar la de un 0,88 entre el recuento de usuarios que piden información y el recuento de usuarios que hacen clic en una web después de hacer una búsqueda transaccional (clics transaccionales). Por último, se ha hecho un análisis de la serie temporal de clics transaccionales y una predicción de 30 semanas con modelos autorregresivos de media móvil.

2010 Mathematics Subject Classification. 62-07, 62-09, 62M10, 60G25

Agradecimientos

Mi agradecimiento mas sincero al equipo de análisis y investigación de mercados del área de marketing de la Universitat Oberta de Catalunya, con especial mención a Raúl Zafra y Francesc Pons, por haberme facilitado las bases de este estudio y por lo mucho que he podido aprender de ellos.

Al tutor de este TFG, el Dr. Josep Vives, por su ayuda y supervisión del trabajo en todas las etapas del desarrollo de este trabajo.

Por último, a mis compañeros, amigos y familiares.

Índice

1. Introducción

"La información existe des de que se creo el universo, pero no le hemos dado valor hasta que hemos aprendido a gestionarla."

En 1962, John W. Tukey habló por primera vez del término "Ciencia de Datos" en su artículo "The Future of Data Analysis"[1], pero hasta el siglo XXI no hemos adquirido la tecnología necesaria para empezar a hablar de Big Data.

En 2001, la empresa norteamericana Gartner definió Big Data como datos que contienen una mayor **variedad** y que se presentan en **volúmenes** crecientes y a una **velocidad** superior. Esto se conoce como "las tres V"[2].

Dicho de otro modo, el Big Data está formado por conjuntos de datos de mayor tamaño y más complejos, especialmente procedentes de nuevas fuentes de datos. Estos conjuntos de datos son tan voluminosos que el software de procesamiento de datos convencional sencillamente no puede gestionarlos. Sin embargo, estos volúmenes masivos de datos pueden utilizarse para abordar problemas empresariales que antes no hubiera sido posible solucionar.

El Big Data conforma tres acciones básicas: Extracción, transformación y análisis. La extracción es la recogida masiva de datos sin estructura aparente y de diferentes fuentes. Una vez recogidos, se deben organizar y se les debe dar forma para transformarlos en bases de datos. Estas deben estar ordenadas para facilitar el trabajo. Por último, solo falta analizar estos datos limpios para sacar resultados que nos puedan servir.

Aplicación del Big Data en marketing digital

Hubo un momento no muy lejano en el que pasearse con una pancarta al aire libre parecía ser lo máximo que una empresa podría hacer en lo que concierne al mundo del marketing. En un avance rápido hasta nuestros días, se puede ver que en marketing se esta analizando cada acción de los consumidores para hacer el mejor uso posible de cada canal. Este ecosistema ha creado un gran interés para los analistas de datos y ha ayudado al crecimiento del Big Data.

El proyecto

En este trabajo se pretende estudiar todo este fenómeno con un proyecto de marketing real donde se aplicarán las tres acciones definidas del Big Data para ver así su potencial.

Por ese motivo, primero se vera la metodología que se usa para hacer la extracción de los datos, a continuación, se explicará el funcionamiento de la creación de bases de datos y la transformación de la información recogida y, por último, la creación de series temporales a partir de las variables creadas. Se va ha estudiar las relaciones entre las series [3] y a analizar su comportamiento tanto actual como futuro.

2. Objetivos

En la actualidad, la mayoría de las empresas que quieren competir en el mundo laboral ya se han pasado a la era digital y se pueden encontrar en diferentes motores de búsqueda. Google, mediante sus diferentes herramientas como *Google Search Console, Google Analitycs o Google Ads* [4], proporciona a las empresas una información muy valiosa sobre la actividad de todos los usuarios de Google que realizan búsquedas relacionadas con la empresa o su producto. Des de las áreas de marketing de las empresas ya se utiliza esta información para conocer a su cliente, mejorar sus campañas o definir sus KPI's[2] pero aún nos quedan años para entender el verdadero potencial de esta información.
El objetivo general de este trabajo es utilizar la información que nos brinda Google junto con los datos internos de la empresa para poder darle una utilidad diferente a la que se le da actualmente. Este objetivo engloba diferentes objetivos mas específicos.

Con la información de Google es posible descubrir patrones de búsqueda [5] y así poder identificar que palabras busca el usuario antes de efectuar un lead[3].

Como se ha comentado anteriormente, en Big Data se suele trabajar con mucho contenido sin formato aparente. En estas condiciones la metodología que se suele seguir es la creación de muchas variables, para después analizar posibles relaciones entre ellas. Puede ser muy interesante utilizar esta metodología para poder encontrar relaciones entre variables aportadas por Google y variables internas como leads o matrículas [6].

La creación de dashboards[4] con nuevos KPI's usando tanto posibles patrones de búsqueda como las variables generadas también es un objetivo importante.

Por último, al estar trabajando con variables que dependen del tiempo, se podrá crear series temporales de estas variables. Entonces, a partir de estas series se hará un análisis exhaustivo para ver su comportamiento y intentar construir un buen modelo de predicción.

[2]KPI: medida del nivel del rendimiento de un proceso. El valor del indicador está directamente relacionado con un objetivo fijado previamente y normalmente se expresa en valores porcentuales.

[3]Lead: acción de rellenar un formulario de solicitud de información en una página web.

[4]Dashboard: tipo de interfaz gráfica de usuario que a menudo proporciona vistas de un vistazo de los indicadores clave de rendimiento relevantes para un objetivo particular o proceso de negocio.

3. Extracción de los datos

El primer paso de este trabajo es la obtención de la información. En este apartado se explicara el proceso de la extracción de los datos y cuales son las fuentes que se utilizaran. La información con la que se trabajará durante este estudio es, en primer lugar, los datos adquiridos de Google sobre la actividad de la página web de la Universitat Oberta de Catalunya (UOC) y, después, datos internos cedidos por la universidad.

En este estudio se utilizará información proporcionada por la Universitat Oberta de Catalunya. Por motivos de confidencialidad para con la empresa tanto algunos datos como algunas visualizaciones serán anonimizados.

3.1. Datos de Google

Google es el motor de búsqueda con más usuarios del mundo, además, si tienes una página web, Google te proporciona información muy importante sobre la actividad de los usuarios en tu página web por medio de sus múltiples herramientas. Para hacer la extracción de esta información utilizaremos dos de sus herramientas: Google Search Console y Google Analitycs.

3.1.1. Google Search Console

Google Search Console es un servicio gratuito para webmasters de Google que permite a los creadores de páginas web comprobar el estado de la indexación de sus sitios en internet por el buscador y optimizar su visibilidad. Con esta herramienta se obtienen las siguientes variables sobre todas las búsquedas de google donde aparece la UOC con frecuencia diaria:

- Query: Combinación de palabras con las que el usuario efectua la búsqueda.
- Page: Url de la página que ha aparecido con la búsqueda.
- Country: Abreviación del país desde el cual se ha hecho la búsqueda.
- Device: Dispositivo desde el cual se ha hecho la búsqueda.
- Clicks: Recuento de veces se ha clicado en la página.
- Impressions: Recuento de veces que se ha visto la página.
- Ctr: Proporción clics por impresiones.
- Position: Posición media en la que aparece en Google.

3.1.2. Google Analytics

Google Analytics es una herramienta de analítica web de la empresa Google lanzada el 14 de noviembre de 2005. Ofrece información agrupada del tráfico que llega a los sitios web según la audiencia, la adquisición, el comportamiento y las conversiones que se llevan a cabo en el sitio web.

Esta herramienta aporta mucha información sobre los usuarios que visitan la página web: sexo, edad... Pero los datos que extraeremos serán las sesiones por día. Esta variable indica el tráfico que recibe una pagina web a diario. Cada vez que entra un usuario en la página web, genera una sesión.

3.2. Datos internos

El principal Objetivo de este estudio es poder sacar más jugo de los datos proporcionados por Google. Por eso es importante tener datos de otras fuentes para enriquecer la base de datos y trabajar con todo el conjunto de información. En este caso se trabajará con datos internos cedidos por la UOC:

- Leads: Recuento de leads* totales, de grado y de máster por día.
- Matrículas: Recuento de matrículas totales, de grado y de máster por día.
- Leads con matrícula: Recuento de leads que acaban en matrícula totales, de grado y de máster por día.
- Tiempo medio entre leads y matrícula
- Información de la competencia: Nombre de todas las universidades de la competencia y sus acrónimos.

4. Transformación de los datos

Una vez se han recogido todos los datos, estos son masivos y tienen diferentes formatos. Es necesario ordenarlos, limpiarlos y estructurarlos para poder trabajar con ellos.

4.1. Creación de la base de datos inicial

Toda esta información que se ha ido extrayendo se tiene que ir guardando y ordenando para poder trabajar con ella, es por eso que se crea una base de datos para almacenar la información utilizando Google Big Query. El servicio web de Google Big Query permite realizar almacenamiento y consulta de conjuntos de datos masivos. Su uso es sencillo y permite estudiar bases de datos casi en tiempo real. En esta primera base de datos se integra toda la información extraída de Search Console, además, utilizando esta información, se crean nuevas variables: A partir de la variable page (url) podemos extraer información valiosa.

La base de datos que se ha creado gracias a la información de Search console tiene la siguiente estructura:

Variable	**Tipologia**	**Descripción**
Query	String	Búsqueda de Google
Page	String	Url de la página
Country	String	Abreviación de país
Device	String	Dispositivo utilizado
Date	Date	Fecha de la búsqueda
CountryName	String	Nombre del país
Clicks	Float	Recuento de veces que se ha clicado
Impressions	Float	Recuento de veces que ha aparecido en una búsqueda
Ctr	Float	Proporción de clics por impresión
Position	Float	Posición media en la que se aparece en Google
Page2	String	Url de la página sin protocolo
Idioma	String	Idioma de la página
Protocolo	String	Protocolo de seguridad
Brand	String	Si la búsqueda lleva la palabra Uoc
Tipologia	String	Tipologia de los estudios
Subtipologia	String	Subtipologia de los estudios
Arees	String	Área de los estudios
Subarees	String	Subarea de los estudios
Tipopag	String	Tipo de página
Titol	String	Nombre de los estudios

Cuadro 1: Base de datos de Big Query

Esta base de datos almacena información de las miles de búsquedas en las que aparece la UOC cada día. Un total de mas de dos años en los que constan mas de 21 millones de

entradas. Por esta razón se utiliza un software potente como es Big Query con el cual se puede manejar gran cantidad de datos utilizando un lenguaje parecido al SQL.

4.2. Transformación de la base de datos

Para llevar a cabo la transformación de los datos de este estudio se ha utilizado R-Studio, un entorno de desarrollo integrado para el lenguaje de programación R, dedicado a la computación estadística y gráficos. Utilizando R-Studio se han elaborado cuatro scripts para trabajar con la base de datos. El primer script simplemente descarga la base de datos de Big Query en formato *.Rdata* para poder trabajar con ellos en R-Studio y hace una copia de seguridad. Los tres importantes son ***KWdata.R, Daydata.R y Weekdata.R***. Cada uno de estos tres scripts tiene la funcion de agregar los datos, crear variables y generar una nueva base de datos. A continuación se ha elaborado un mapa conceptual para entender mejor el proceso.

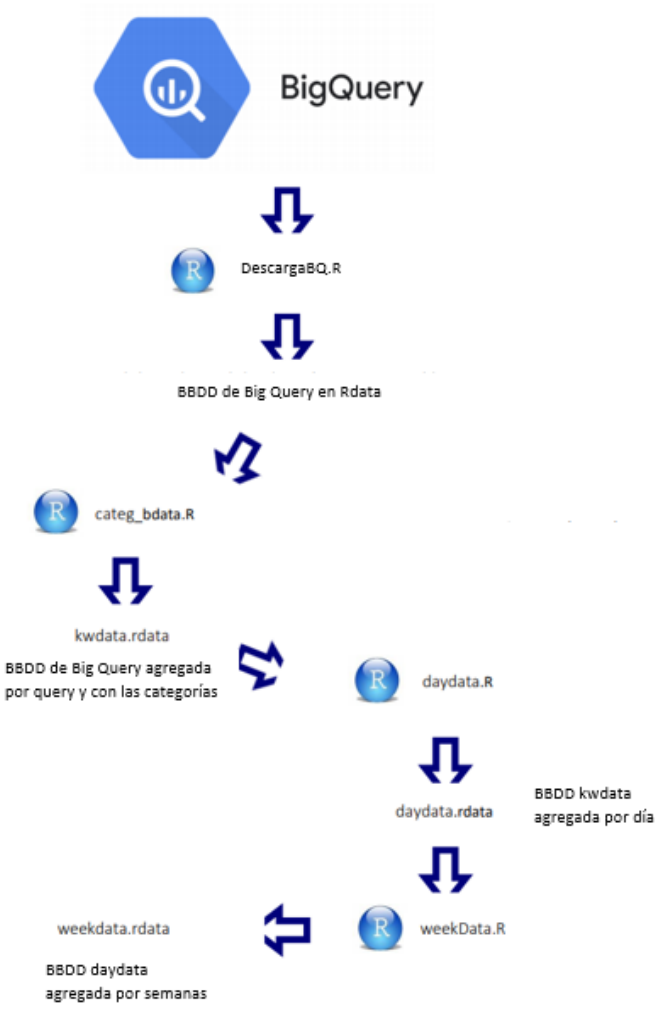

Figura 1: Mapa conceptual del proceso de transformación

4.2.1. Categorización de las búsquedas (KWdata.R)

La función principal de este script es utilizar un algoritmo que analizará y clasificará las búsquedas según la intencionalidad que tiene el usuario al realizarla. Para ejecutar KWdata.r ha sido necesario optimizar mucho el script y utilizar un ordenador potente dado el tamaño de la base de datos. En esta transformación se hará un agregado por query y pasaremos de trabajar con una base de 21 millones de entradas a una con más de 150.000. Por razones de confidencialidad con la empresa esta clasificación no será revelada.

Las categorías generadas son las siguientes:

- Las **informacionales simples** son búsquedas donde su objetivo es informarse y lo hacen formulando una pregunta. Por ejemplo: ¿Dónde puedo estudiar Ade?
- Las **informacionales normales** son búsquedas que también tienen como objetivo informarse pero lo hacen sin formular pregunta. Por ejemplo: Estudiar Ade en Barcelona.
- Las **navegacionales** son búsquedas que quieren información más dirigida, llevan el nombre de la marca. En esta categoría nos hemos centrado en el nombre de marca de la competencia. Por ejemplo: Estudiar Ade en Uned.
- Las **navegacionales de marca** son como las anteriores pero en este caso se ha centrado en las búsquedas que lleven la marca UOC. Por ejemplo: Estudiar ade en la UOC.
- Las **compromiso** són búsquedas que van más atadas a alumnos o que tienen un compromiso con la marca. Por ejemplo: Convalidaciones Uoc ade, beca psicologia Uned, acceso campus.
- Las **transaccionales** son las búsquedas de precios o descuentos, por lo tanto, las que más se acercan al momento de la compra. Por ejemplo: Precios Ade, matrícula UOC, precio crédito Uned.

4.2.2. Creación de base de datos por fecha (Daydata)

Con el script daydata.r se pretende crear una base de datos para poder hacer un análisis de series temporales por lo que solo se utilizaran las variables numéricas y la variable creada categoría. Para llevarlo a cabo se hace un agregado por fecha de todos los datos de KWdata de manera que se consigue tener la información con frecuencia diaria. Para hacer el agregado se usará el sumatorio para los clics y las impresiones, y la media para el CTR y la posición. Se calcula la frecuencia de queries para conseguir la variable *recuento*, que da el número de búsquedas que se hacen cada día. Hacemos lo mismo para la variable categoría de manera que se dividirá en seis variables, una para cada categoría, y así logramos que después de agregar con sumatorio tendremos el recuento de búsqueda de cada categoría por día.

Una vez creada esta nueva base de datos por día, se añaden por un lado datos conseguidos de la extracción como las sesiones, los leads, los leads matriculados, las matrículas y el tiempo medio entre lead y matrícula.

Por otro lado, se crean nuevas variables para enriquecer la base. Se crean las variables de clics por cada categoría de manera que se podrá saber, por ejemplo, el recuento de clics que se han hecho haciendo una búsqueda transaccional. Se hace lo mismo con las impresiones.

4.2.3. Eliminación de ruido (Weekdata.R)

Cuando se trata de desarrollar series temporales con esta clase de datos, uno se encuentras con que el flujo de usuarios navegando es muy diferente dependiendo del día de la semana. Se tiene un gran flujo entre semana que desciende al llegar el sábado y domingo. Para reducir el ruido que esto genera se desarrolla este último script.

La funcionalidad de Weekdata.R es acabar con este ruido haciendo una agregado por semanas de manera que tendremos toda la información de Daydata pero por semanas. Weekdata.R también lidia con los típicos problemas de que no todos los años empiezan con el mismo día de la semana o que no tienen el mismo número de días.

5. Análisis de los datos

En un primer momento se ha hecho una extracción masiva de datos desde distintas fuentes, a continuación, se ha filtrado esta información ordenándola de diversas maneras y se han creado nuevos datos. Por último, toca analizar toda esta información con distintos conceptos matemáticos.

5.1. Correlación

5.1.1. Definición

El primer concepto en el se ha trabajado durante el análisis de los datos es la correlación. La correlación es una medida de dependencia lineal entre dos variables aleatorias cuantitativas. Esta tiene la cualidad de permitir conocer tanto la intensidad como la dirección de la relación existente entre dichas variables. Así pues, dadas dos variables x e y con covarianza,

$$\sigma_{xy} = E[(y - \bar{y}) \cdot (x - \bar{x})], \tag{5.1}$$

el coeficiente de correlación de Pearson se define como

$$r_{xy} = \frac{\sigma_{xy}}{\sigma_x \cdot \sigma_y}, \tag{5.2}$$

donde σ_x y σ_y son las desviaciones típicas y se definen como

$$\sigma_x = \sqrt{E[(x - \bar{x})^2]} \quad y \quad \sigma_y = \sqrt{E[(y - \bar{y})^2]}. \tag{5.3}$$

El coeficiente de correlación de Pearson es un indicador que define el grado de dependencia lineal que hay entre dos variables. Este valor varía en el intervalo $[-1, 1]$. Cuanto más próximo al 1 más correlacionadas estan las variables, cuanto más cerca del 0 menos menos correlacionado y si se acerca al -1 significa que las dos variables están inversamente correlacionadas.

La correlación entre dos series temporales, tanto en tiempo simultáneo, como desfasadas en el tiempo, se conoce como correlación cruzada. Sean X_t y Y_t dos series temporales donde $t = 0, 1, \cdots, N$.

El coeficiente de correlación cruzada entre las dos series se calcula:

$$r_{XY} = \frac{\sum_{t=0}^{N} E[(Y_t - \bar{Y}) \cdot (X_t - \bar{X})]}{N \cdot \sigma_X \cdot \sigma_Y} \tag{5.4}$$

donde $\bar{X}, \bar{Y}$ son las medias y σ_X, σ_Y la desviación estándar de las series.

5.1.2. Análisis

Dado que este estudio tenemos una gran cantidad de variables, la manera más rápida y visual de ver la correlación entre todas las variables es con una matriz de correlaciones. Llamaremos matriz de correlación a la matriz simétrica que tiene unos en la diagonal y fuera de ella los coeficientes de correlación entre las variables. Escribiremos

$$R = \begin{pmatrix} 1 & \mathrm{r}_{12} & \cdots & \mathrm{r}_{1k} \\ \vdots & \vdots & \ddots & \dots \\ \mathrm{r}_{k1} & \mathrm{r}_{k2} & \cdots & 1 \end{pmatrix} \tag{5.5}$$

Sea r_{ij} una componente de la matriz, esta indica el coeficiente de correlación entre la variable i y la j.
Una vez creada la matriz con el paquete *Corrplot* [7] de Rstudio y quitando las variables que no dan información relevante, queda de la siguiente manera:

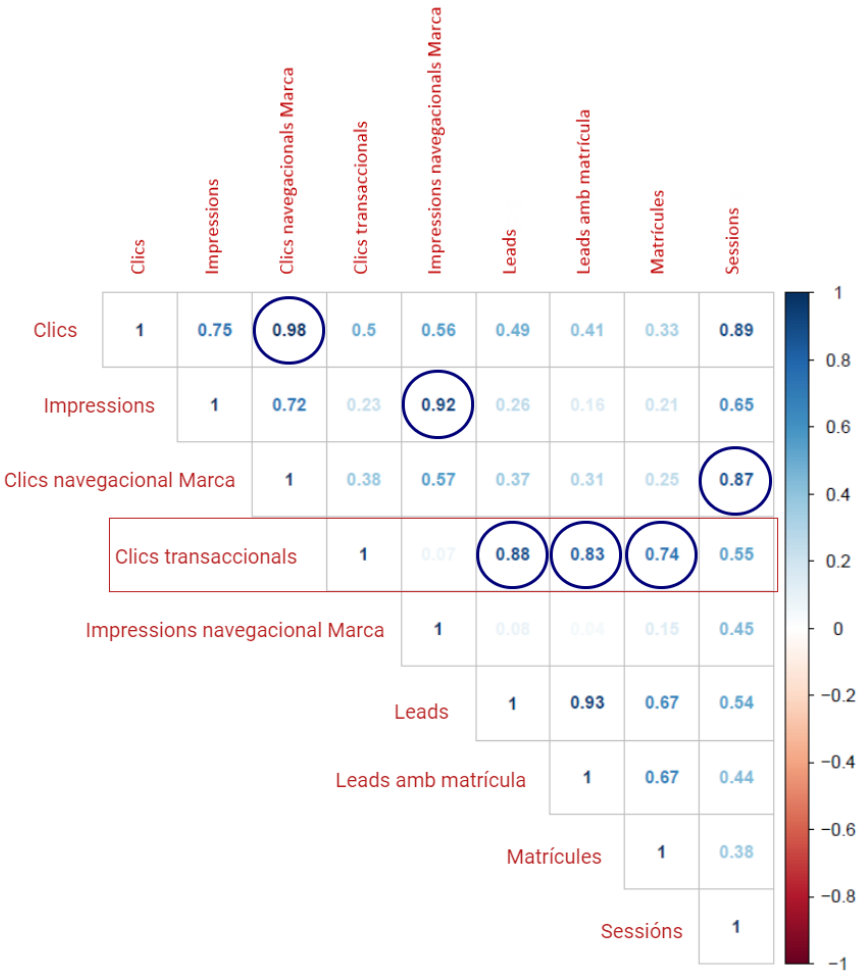

Figura 2: Matriz de correlaciones

Una vez obtenidos estos resultados, podemos sacar distintas conclusiones. Encontramos que los clics están fuertemente relacionados con los clics navegacionales de marca (Figura 3), este hecho era esperado dado que los clics navegacionales representan un 79 % del total de los clics. En el caso de las impresiones (Figura 4), las impresiones navegacionales de marca son el 74 % de las totales, así que también explicaría la correlación tan alta.

Por motivo de confidencialidad y para anonimizar la información, las series temporales con las que se trabajará están sin el eje de las ordenadas y con el eje temporal desplazado.

Aun así, es interesante observar gráficamente el significado de una correlación tan alta entre dos series temporales:

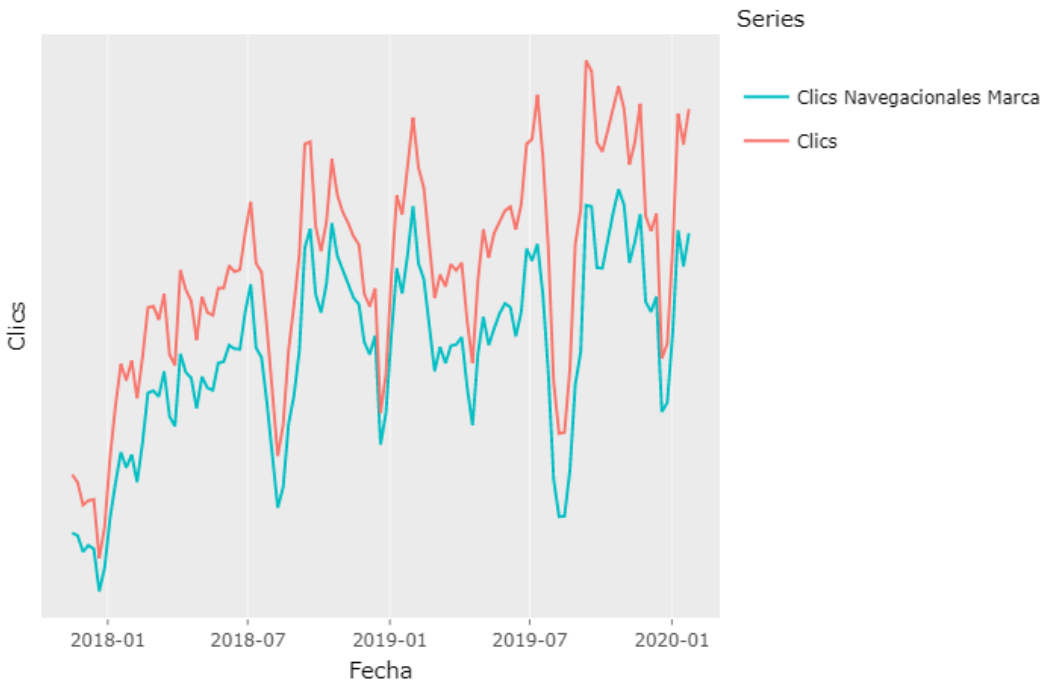

Figura 3: Comparativa entre clics y clics navegacionales de marca

En la figura 3 se puede apreciar la relación entre los clics y los clics navegacionales de marca con un coeficiente de correlación de 0,98.

En la figura 4 se puede apreciar la relación entre las impresiones y las impresiones navegacionales de marca con un coeficiente de correlación de 0,92. Si se normalizaran las cifras, tanto en la figura 3 como en la figura 4, gráficamente serían prácticamente las mismas.

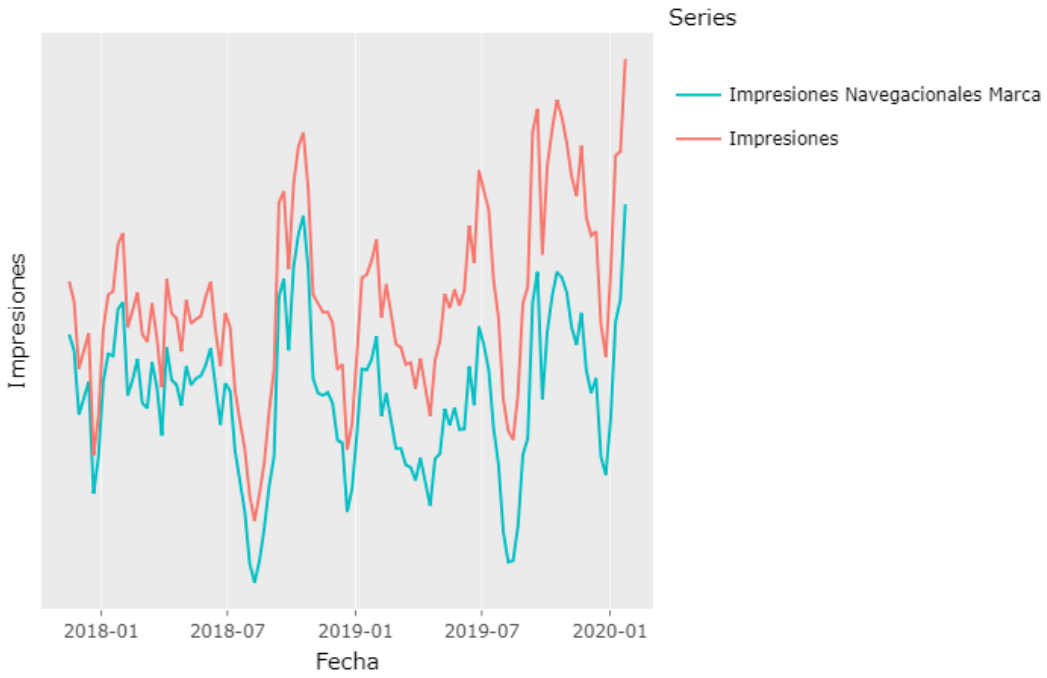

Figura 4: Comparativa entre impresiones y impresiones navegacionales de marca

Se puede observar que las sesiones y los clics navegacionales de marca también estan fuertemente relacionado (Figura 5), sería también interesante estudiar este comportamiento.

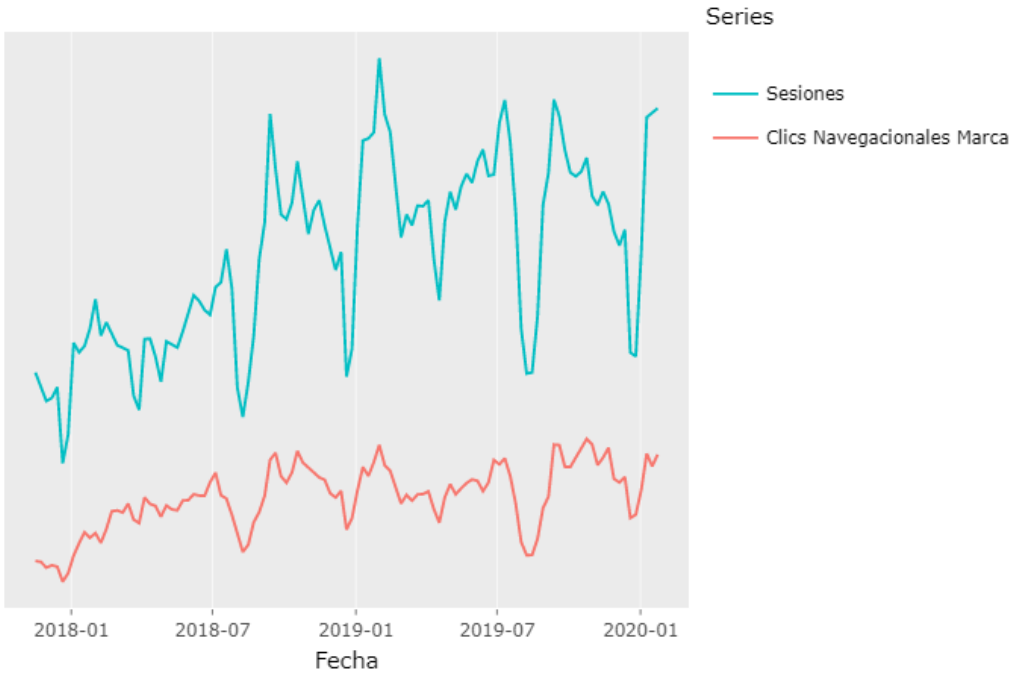

Figura 5: Comparativa entre sesiones y clics navegacionales de marca

En la figura 5 se puede apreciar que la relación entre las variables no es tan fuerte como

en las figuras 3 y 4. Aun así sigue teniendo un significativo coeficiente de correlación de 0,87.

Por último y en lo que más se va a fijar este estudio, las relaciones con los clics transaccionales.

En la figura 6 se puede observar la comparación de los clics transaccionales con los leads que hemos visto que tienen una correlación del 0,88. Aunque las cifras sean tan diferentes eso no hace variar el indice de correlación dado que este no depende tamaño.

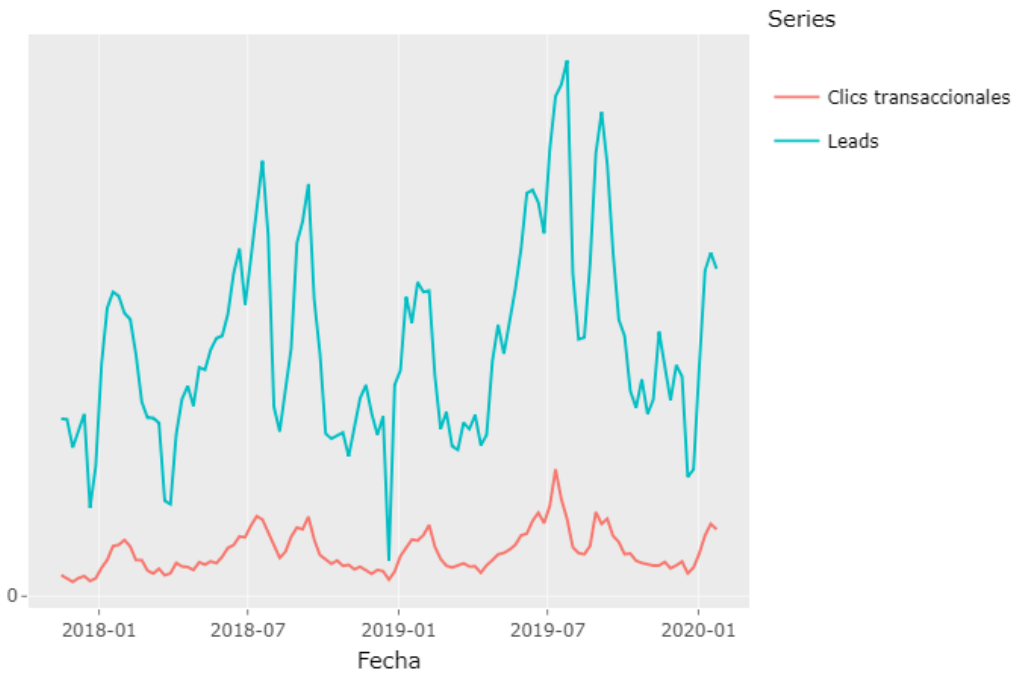

Figura 6: Comparativa entre clics transaccionales y leads

En la figura 7 se puede observar perfectamente la correlación del 0,83 entre las dos series temporales. Estos resultados son muy satisfactorios dado que podemos establecer como indicador de compra para un negocio el recuento de la gente que hace clic en una página web después de realizar una búsqueda transaccional.

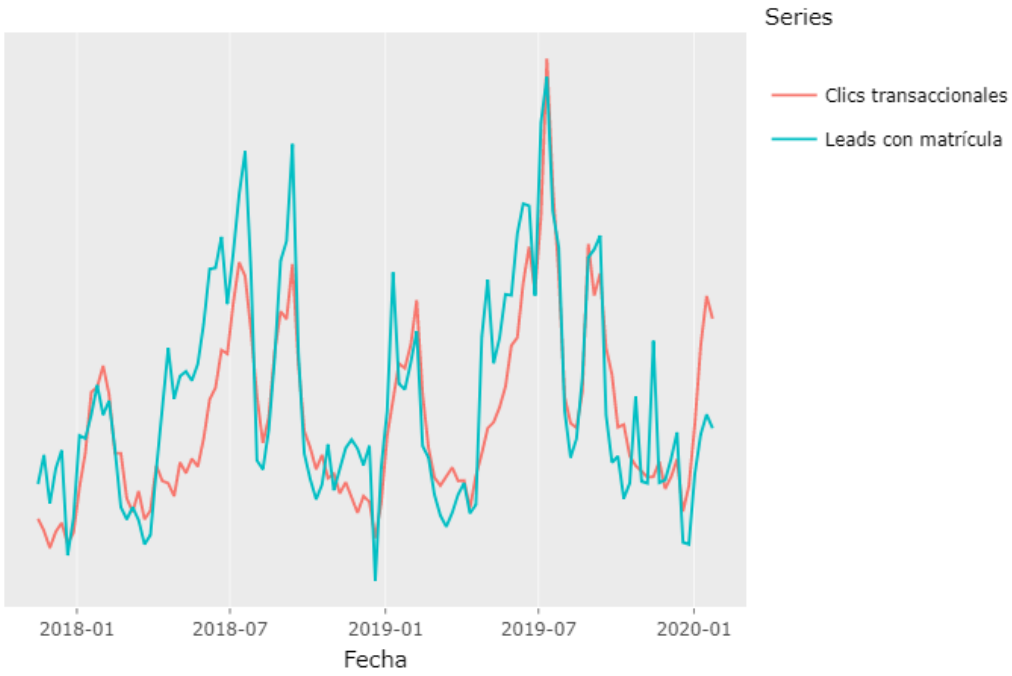

Figura 7: Comparativa entre clics transaccionales y leads con matrícula

Por último, la correlación en la figura 8 baja hasta un 0,74. Esto hecho es normal dado que cuando la gente se quiere matricular primero hace pide información (lead) días mas tarde se matricula. Por este motivo se creo la variable leads con matrícula, es una método mas acertado para observar la relación de los clics transaccionales y las matrículas.

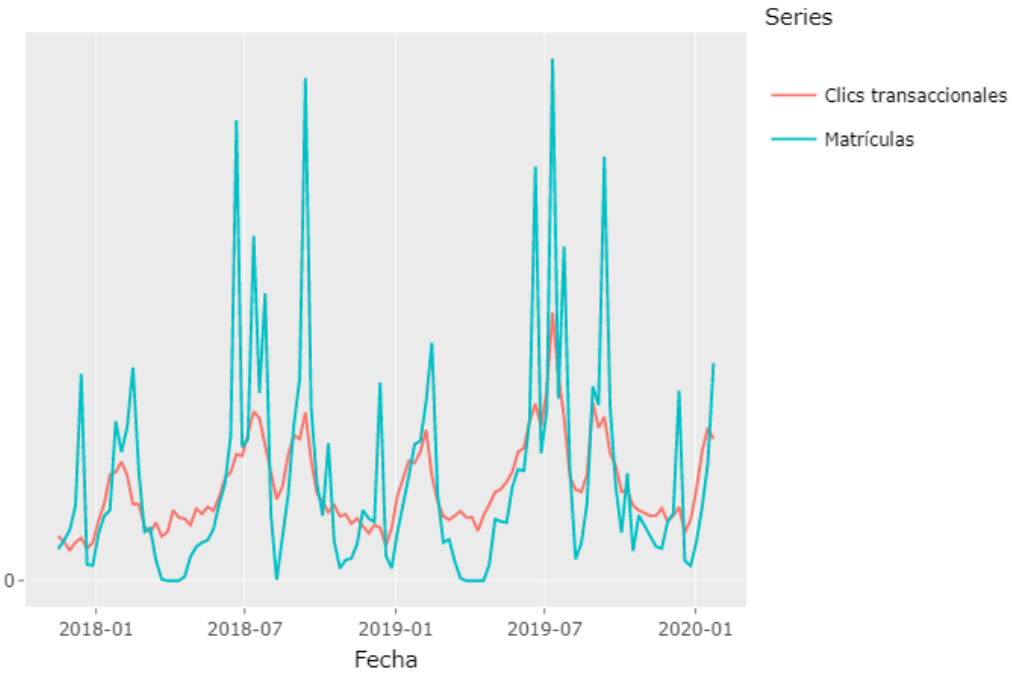

Figura 8: Comparativa entre clics transaccionales y matrículas

El principal objetivo de este trabajo es darle otra utilidad a los valores de marketing que aporta Google. Dada la asociación que observamos entre los clics transaccionales con los leads, leads con matrículas y matrículas, se puede intentar encontrar esta nueva utilidad. Como la información de leads, leads con matrículas y matrículas es de suma importancia, puede ser muy interesante crear la serie temporal de de los clics transaccionales y posteriormente hacer un análisis exhaustivo.

5.2. Series temporales

Las principales fuentes consultadas para la elaboración de este tema son las obras [8], [9] para la teoría y [10] para el análisis.

5.2.1. Definición

Una serie temporal se la define como un conjunto de mediciones que describen la evolución de un fenómeno o de una variable a lo largo del tiempo.

De manera formal, es una secuencia cronológicamente ordenada de valores de medición sobre el estado de una variable cuantitativa de un fenómeno o proceso. Dichas mediciones están ordenadas respecto al tiempo y son generalmente dependientes entre sí. Esta dependencia entre las observaciones jugará un papel importante en el análisis de la serie.

De esta manera, una serie temporal puede representar distintos fenómenos, desde temperaturas, precios, poblaciones, hasta visitas de un página web o matrículas en un máster. Las observaciones de una serie temporal suelen ser denotadas como:

$$Y_1, Y_2, Y_3, \cdots, Y_{t-1}, Y_t$$

donde Y_t es el valor de la serie en el instante t.

Como se menciono anteriormente, las series temporales están ordenadas respecto al tiempo y es por eso que se puede distinguir entre series de **alta frecuencia** y series de **baja frecuencia**. Cuando los datos son recogidos anual, trimestral o hasta mensualmente, se consideran de baja frecuencia, mientras que si se recogen semanalmente, diariamente o por horas, se habla de alta frecuencia.

Cuando se quiere hacer un estudio descriptivo de una serie temporal tenemos que descomponer la variación de una serie en varias componentes básicas. Este enfoque no tiene porque ser el más adecuado, pero es interesante utilizarlo cuando en la serie se observa una tendencia o periodicidad. Este enfoque descriptivo consiste en encontrar componentes que correspondan a una tendencia a largo plazo, un comportamiento estacional y una componente aleatoria o residuo.

Las componentes o fuentes de variación que se consideran habitualmente son las siguientes:

1. **Tendencia:** Se puede definir como un cambio a largo plazo que se produce en relación al nivel medio, o el cambio a largo plazo de la media. La tendencia se identifica

con un movimiento suave de la serie a largo plazo.

2. **Efecto Estacional:** Muchas series temporales presentan cierta periodicidad o dicho de otro modo, variación de cierto periodo (anual, mensual ...). Por ejemplo, el paro laboral aumenta en general en invierno y disminuye en verano. Estos tipos de efectos son fáciles de entender y se pueden medir explícitamente o incluso se pueden eliminar del conjunto de los datos, desestacionalizando la serie original.

3. **Componente Aleatoria:** Una vez identificados los componentes anteriores y después de haberlos eliminado, persisten unos valores que son aleatorios. Se pretende estudiar qué tipo de comportamiento aleatorio presentan estos residuos, utilizando algún tipo de modelo probabilístico que los describa.

Entonces podemos denotar una serie temporal como el proceso:

$$X_t = T_t + E_t + A_t$$

Donde T define la tendencia , E el efecto estacional y A la componente aleatoria en el instante t.

5.2.2. Análisis

Tras haber analizado la matriz de correlación y obtener buenos resultados de algunas variables, se ha visto la importancia de hacer un buen análisis de la variable de clics transaccionales. Recordar que esta variable mide el recuento de usuarios que clican en nuestra página web después de haber hecho una búsqueda de tipo transaccional.

Por motivo de confidencialidad y para anonimizar la información, las series temporales con las que se trabajará están sin el eje de las ordenadas y con el eje temporal desplazado.

En primer lugar se procede a graficar la serie temporal de la variable de clics transaccionales con frecuencia diaria (Figura 9).

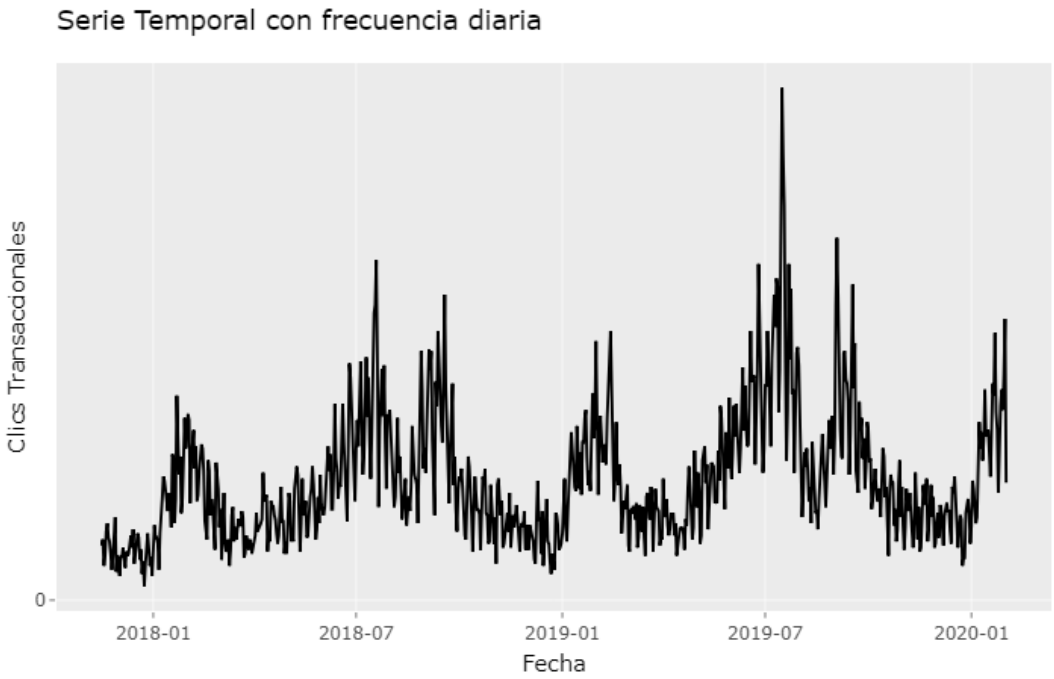

Figura 9: Serie temporal con frecuencia diaria

Como se ha explicado anteriormente, el tráfico en Google cae los fines de semanas, ese hecho genera mucho ruido en mediciones con frecuencia diaria que si no se quita puede provocar errores en el análisis. Para eliminar ese ruido se ha optado por utilizar una frecuencia semanal y se han agregado los clics en forma de sumatorio.
Todo esto se trabaja en el script Weekdata.r y los resultados se pueden observar en la Figura 10:

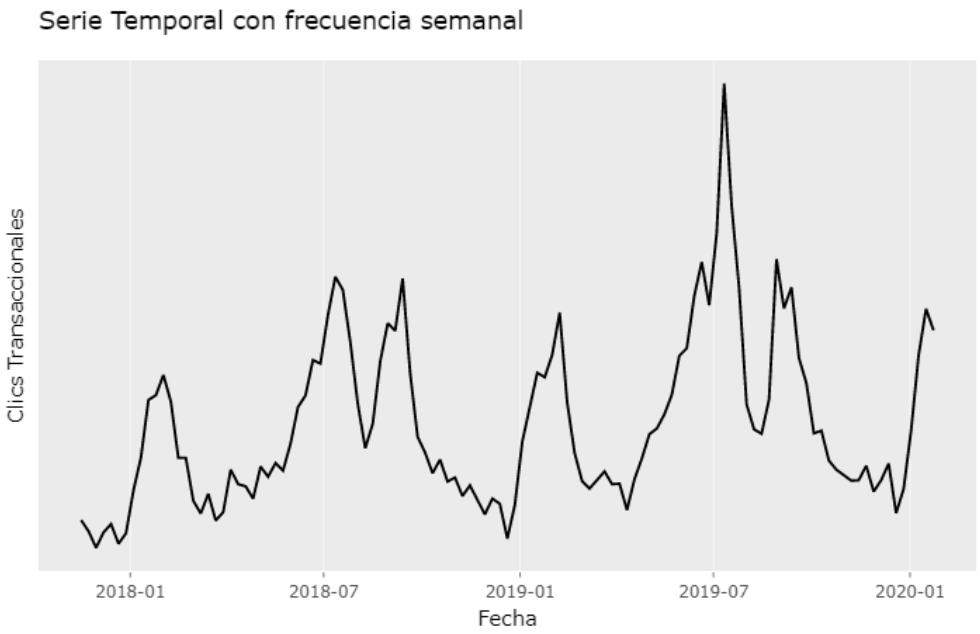

Figura 10: Serie temporal con frecuencia semanal.

Una vez construida nuestra serie temporal limpia de ruido se debe encontrar la tendencia linear de esta. Para ello buscamos la recta de regresión lineal de los puntos de manera que nos muestra la tendencia.
La recta encontrada es:

$$Y = 1{,}75 \cdot X + 315{,}39$$

Esto indica que la tendencia es positiva y por lo tanto, según se avanza en el tiempo, más gente hace clic en la página web.

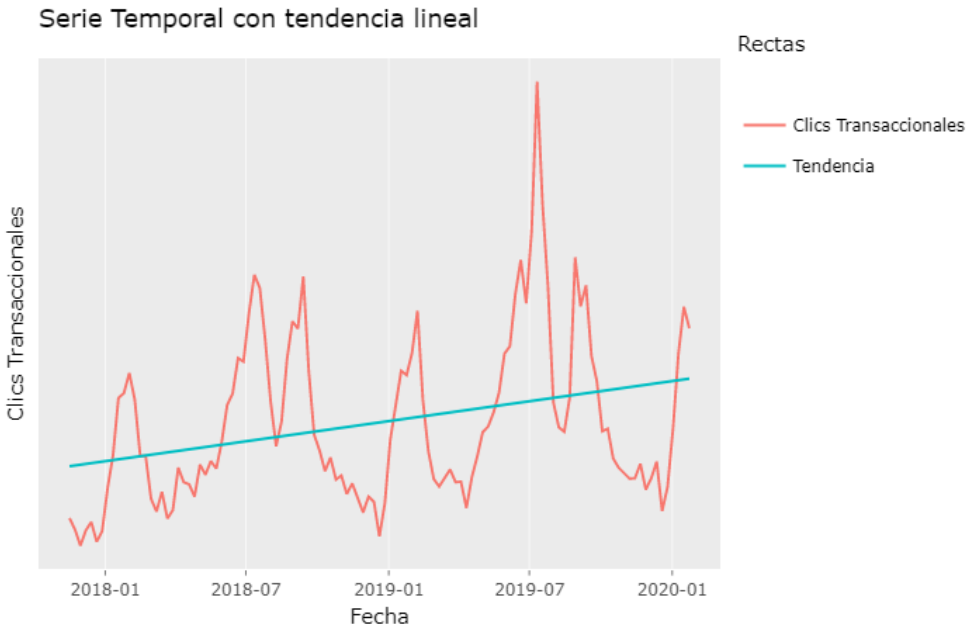

Figura 11: serie temporal con tendencia lineal.

Una vez encontrada la tendencia (Figura 11), para poder proseguir con el análisis, debemos quitarle la tendencia a los valores observados de manera que nos quede una serie temporal con tendencia 0 (Figura 12).

Sea T_t la variable tendencia para todo instante t y X_t el valor de la serie temporal en el instante t, entonces $X'_t = X_t - T_t$ será la serie temporal sin tendencia. Gráficamente se verá así:

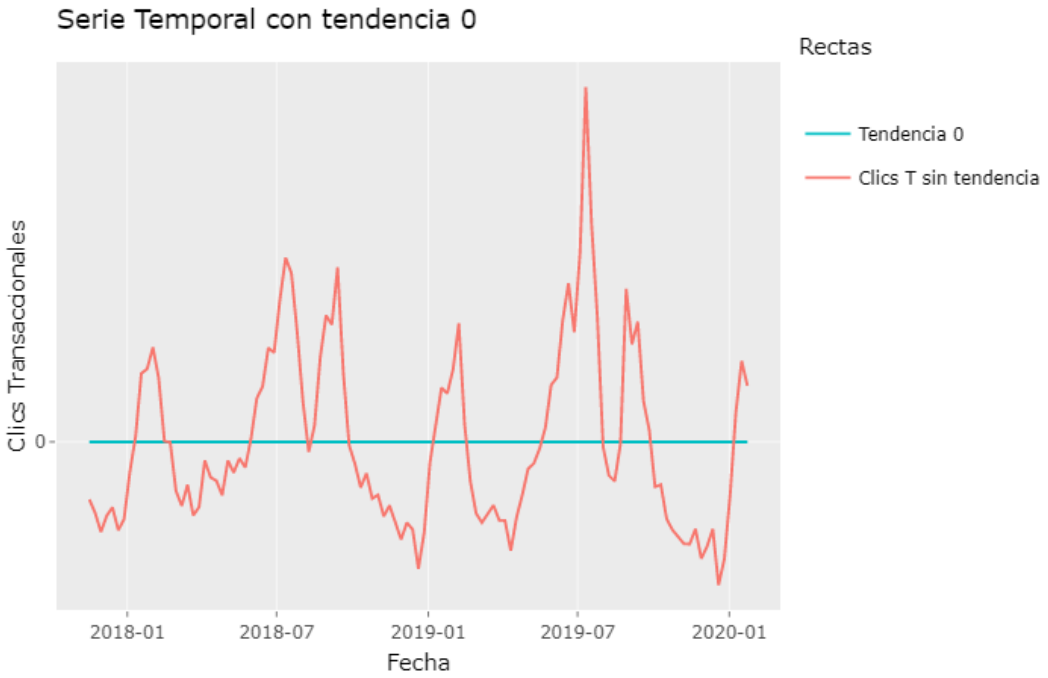

Figura 12: serie temporal con tendencia lineal 0.

Llegados a este punto debemos encontrar el efecto estacional E_t. Hay diversos métodos para encontrar ciclos. En este caso visualmente se puede apreciar que coger como ciclo un año es lo mas óptimo dado que estamos hablando de una universidad y los acontecimiento se repiten anualmente. Para estimar el efecto estacionario dividimos el año en 52 semanas, ya que la información esta recogida de esta manera, y calculamos las medias aritméticas de clics transaccionales de cada semana.

Sea j la semana del año, n el recuento de años que se ha recogido datos esa semana y $X_{i,j}$ el recuento de clics transaccionales en la semana j del año i. Los valores de la componente estacionaria se calculan de la siguiente manera:

$$E_j = \sum_{i=1}^{n} \frac{X_{i,j}}{n} \qquad \forall j \in \{1, 2, \cdots, 52\} \tag{5.6}$$

Si replicamos el patrón y lo extendemos para toda la muestra encontraremos la componente estacionaria de la serie temporal (Figura 13). Al dibujarla queda la siguiente serie:

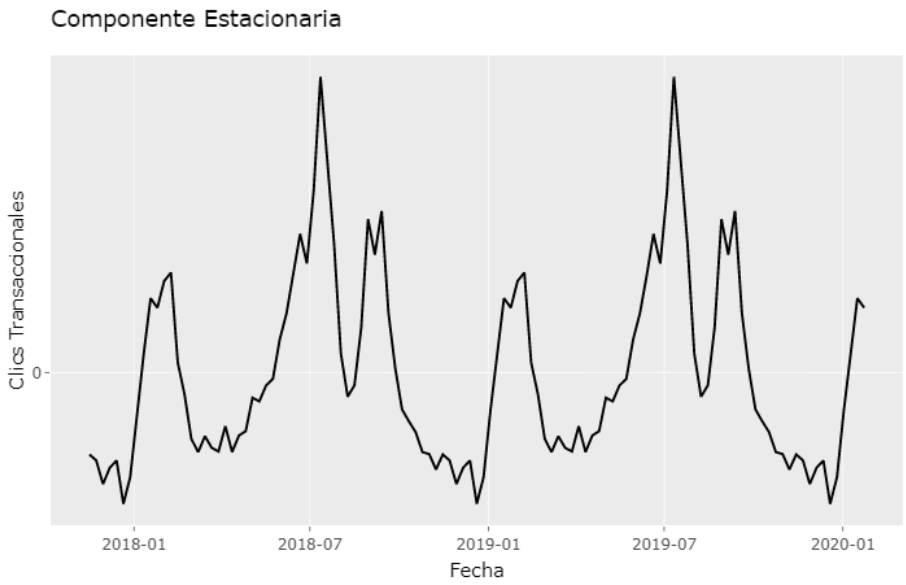

Figura 13: Componente estacional de la serie temporal.

Se puede observar que la gráfica (Figura 13) mantiene los picos de enero, julio y setiembre. Si comparamos el gráfico de los valores observados con tendencia 0 y la componente estacionaria anterior, se tiene un gráfico (Figura 14) muy interesante donde se pueden observar claramente las diferencias.

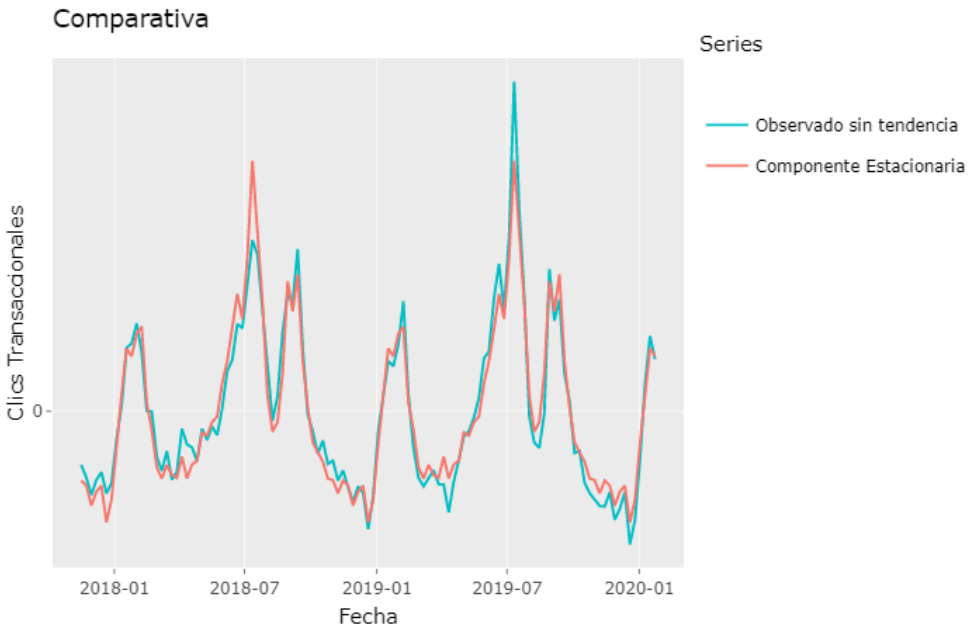

Figura 14: Comparativa de la serie temporal con su componente estacionaria.

Para acabar con la descomposición de la serie temporal, se encuentra la componente aleatoria. Para ello, calculamos la diferencia entre los valores observados con tendencia 0 y la componente estacionaria.

$$A_t = (X_t - T_t) - E_t \tag{5.7}$$

El resultado es el siguiente gráfico de barras:

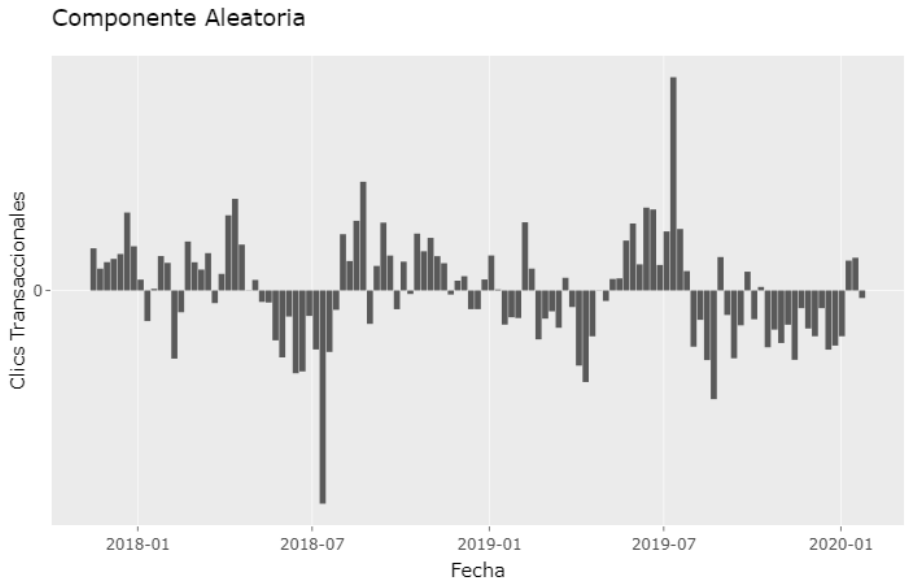

Figura 15: Componente aleatoria de la serie temporal.

En la figura 15 se puede observar que la componente aleatoria parece aproximadamente estacionaria y sin patrones de tendencia o estacionalidad. La variable es bastante estable exceptuando en julio. Como hay tanta diferencia entre el pico de 2018 y el de 2019 eso hace crecer la componente aleatoria. Si se tuviera una muestra mayor, este acontecimiento no se daría.
En el siguiente gráfico se puede observar mejor las dimensiones de la componente aleatoria al juntarla con la componente estacionaria:

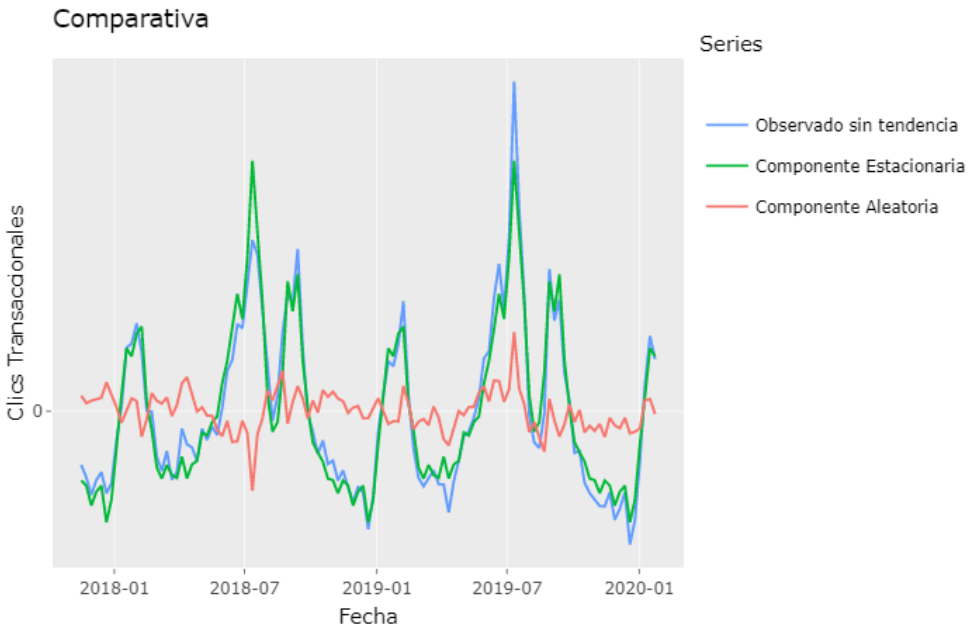

Figura 16: Comparativa de la componente aleatoria y estacional con la serie temporal.

5.3. Métodos predictivos

Las principales fuentes consultadas para la elaboración de este tema son las obras [11], [12] para la teoría y [13], [14] para el análisis.

En 1976, G. E. P. Box y G. M. Jenkins, publicaron su obra: "Time Series Analysis: Forecasting and Control"[11]. Este hecho estableció un punto de inflexión en las técnicas cuantitativas de predicción. La metodología propuesta por estos autores, trata de realizar previsiones acerca de los valores futuros de una variable utilizando únicamente como información, la contenida en los valores pasados de la propia serie temporal. Este enfoque supone una alternativa a la construcción de modelos uniecuacionales o de ecuaciones simultáneas, pues supone admitir que las series temporales poseen un carácter estocástico, lo que implica que deben analizarse sus propiedades probabilísticas.

En este estudio, se trabajará con una parte de la metodología ARIMA, los modelos autorregresivos de media móvil.

5.3.1. Procesos estocásticos

Podemos definir un proceso estocástico como un conjunto de variables aleatorias asociadas a distintos instantes del tiempo. Así, en cada período o momento temporal, se dispone de una variable que tendrá su correspondiente distribución de probabilidad; por ejemplo, si consideramos el proceso Y_t, para $t = 1$, tendremos una variable aleatoria, Y_1, que tomará diferentes valores con diferentes probabilidades.

La relación existente, por tanto, entre una serie temporal y el proceso estocástico que la genera es similar a la que existe entre una muestra y la población de la que procede, de tal forma que podemos considerar una serie temporal como una muestra o realización de un proceso estocástico, formada por una sola observación de cada una de las variables que componen el proceso. En el estudio se tratará deducir la forma del proceso estocástico a partir de las series temporales que genera.

Un proceso estocástico, X_t, se suele describir mediante las siguientes características: esperanza matemática, varianza, autocovarianzas y coeficientes de autocorrelación.

La esperanza matemática de X_t se traduce en la sucesión de las esperanzas matemáticas de las variables que componen el proceso a lo largo del tiempo, tal que:

$$E(X_t) = \mu_t, \qquad t = 1, 2, 3, ..., n \tag{5.8}$$

Por su parte, la varianza de un proceso aleatorio es una sucesión de varianzas, una por cada variable del proceso:

$$V(X_t) = E(X_t - \mu_t)^2, \qquad t = 1, 2, 3, ..., n \tag{5.9}$$

Las autocovarianzas, por su parte, son las covarianzas entre cada par de variables del proceso, tales que:

$$C(X_t, X_{t+k}) = E[(X_t - \mu_t)(X_{t+k} - \mu_{t+k})], \qquad t = 1, 2, 3, ..., n \tag{5.10}$$

Finalmente, los coeficientes de autocorrelación son los coeficientes de correlación lineal entre cada par de variables que componen el proceso:

$$r_k = \frac{C(X_t, X_{t+k})}{\sqrt{V(X_t)}\sqrt{V(X_{t+k})}}, \qquad t = 1, 2, 3, ..., n \tag{5.11}$$

donde $-1 \leq r_k \leq 1$.

Por último, a partir de los coeficientes de autocorrelación, vamos a definir dos funciones que usaremos durante el análisis:

- Por un lado, la función de autocorrelación simple (ACF) o correlograma, la cual es la representación gráfica de los coeficientes de autocorrelación en función de los distintos retardos o desfases entre las variables.

- La función de autocorrelación parcial (PACF), que mide la correlación existente entre dos variables del proceso en distintos períodos de tiempo, pero una vez eliminados los efectos sobre las mismas de los períodos intermedios. Por ejemplo, puede que exista cierta correlación entre Y_t e Y_{t-2}, debido a que ambas variables estén correlacionadas con Y_{t-1}.

5.3.2. Procesos estacionarios

Como se ha comentado anteriormente, un proceso estocástico es estacionario si todas las variables aleatorias que lo componen están idénticamente distribuidas, independientemente del instante del tiempo en que se estudie el proceso.

Definición 5.1. *Un proceso estocástico* $\{X_k, k \in \mathbb{Z}\}$ *es estrictamente estacionario si para cualquier* $k_1, \cdots, k_n$ *y* l*, los vectores*

$$(X_{k_1}, \cdots, X_{k_n}) \tag{5.12}$$

y

$$(X_{k_1+l}, \cdots, X_{k_n+l}) \tag{5.13}$$

tienen la misma ley.

Sin embargo, esta es la versión más estricta de la estacionariedad de un proceso. En la práctica pocas veces se puede utilizar, así que por lo general, se usa un concepto menos exigente, el de estacionariedad débil. La estacionariedad débil se da cuando la media del proceso es constante e independiente del tiempo, la varianza es finita y constante, y el valor de la covarianza entre dos periodos depende únicamente de la distancia o desfase entre ellos, sin importar el momento del tiempo en el cual se calculan. Es importante cuando una serie es no estacionaria investigar si es no estacionaria en media o en varianza.

La razón fundamental por la cual es tan importante que el proceso analizado sea estacionario, es que los modelos de predicción de series temporales que veremos a continuación están diseñados para ser utilizados con procesos de este tipo. Si las características del proceso cambian a lo largo del tiempo, resultará difícil representar la serie para intervalos de tiempo pasados y futuros mediante un modelo lineal sencillo, no pudiéndose por tanto realizar previsiones fiables para la variable en estudio.

Por regla general, las series con las que tratamos no suelen ser de procesos estacionarios, sino que suelen tener una tendencia, ya sea creciente o decreciente, y variabilidad no constante. Dicha limitación en la práctica no es tan importante porque las series no estacionarias se pueden transformar en otras aproximadamente estacionarias. El caso en el que se esta trabajando se ha efectuado esta transformación quitándole a la serie la tendencia y la componente estacional.

Ejemplos:

Definición 5.2. *Ruido blanco: Se dice que* $X_k, k \geq 1$ *es un ruido blanco si todas las variables tienen esperanza* μ *, varianza* σ^2 *y no están correlacionadas.*

Definición 5.3. *Ruido IID: Se dice que* $X_k, k \geq 1$ *es un ruido IID si las variables aleatorias son independientes e idénticamente distribuidas con media* μ *y desviación estándar* σ*. El ruido IID es un caso de ruido blanco dado que independencia implica no correlación.*

5.3.3. ARMA

Modelos autorregresivos AR(p)

Los procesos autorregresivos son aquellos que representan los valores de una variable durante un instante del tiempo en función de sus valores precedentes. Así, un proceso autorregresivo de orden p, AR(p), tendrá la siguiente forma:

$$X_t = \delta + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + e_t \tag{5.14}$$

donde δ, es un término constante y e_t es un ruido blanco, que representa los errores del ajuste y otorga el carácter aleatorio al proceso.

Modelos de media móvil MA(q)

En los procesos de media móvil de orden q, cada observación Y_t es generada por una media ponderada de perturbaciones aleatorias con un retardo de q períodos, tal que:

$$X_t = \delta + e_t - \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q} \tag{5.15}$$

donde e_t es un ruido blanco.

Modelos mixtos ARMA(p, q)

Los procesos ARMA (p, q) son, como su nombre indica, un modelo mixto que posee una parte autorregresiva y otra de media móvil, donde p es el orden de la parte autorregresiva y q el de la media móvil. La expresión genérica de este tipo de procesos es:

$$X_t = \delta + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + e_t - \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-q} \tag{5.16}$$

5.3.4. Análisis

Una vez introducidos estos conceptos, procederemos a utilizarlos para la modelización de la serie temporal. Para ello, primero veremos si la serie temporal de clics transaccionales es estacionaria o no estacionaria.

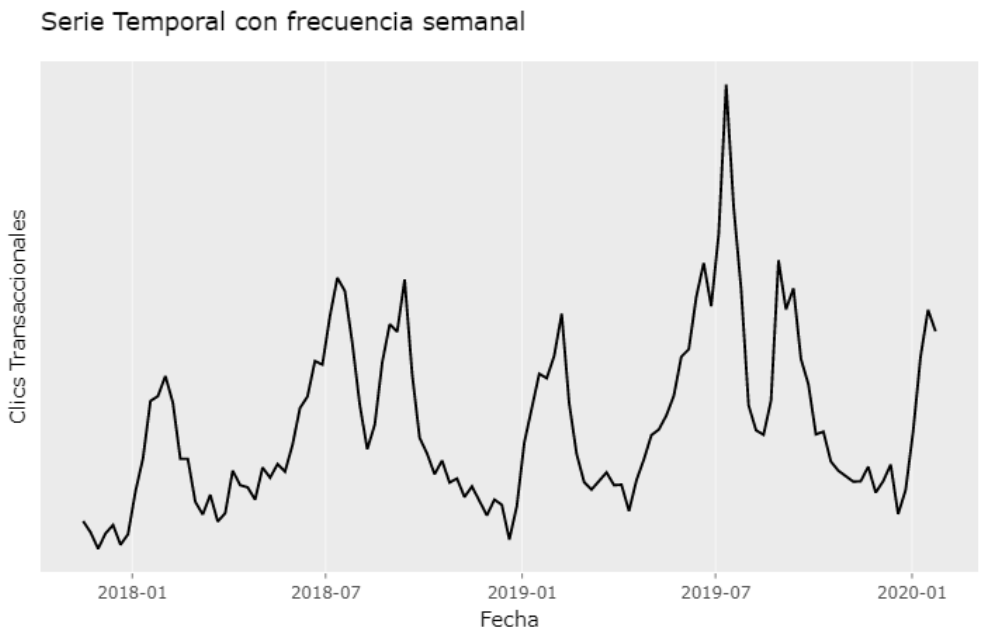

Figura 17: Serie temporal con frecuencia semanal.

Tal y como se observa en la figura 17, gráficamente presenta una tendencia y una estacionalidad que son buenos indicadores de no estacionaridad. Como se ha explicado, los modelos de predicción de series temporales que hemos visto, están diseñados para ser utilizados con procesos estacionarios. Por ese motivo se debe transformar nuestra serie. Primero se debe quitar la tendencia T_t y, posteriormente, la componente estacional E_t. Como resultado se obtiene la componente aleatoria A_t (figura 18).

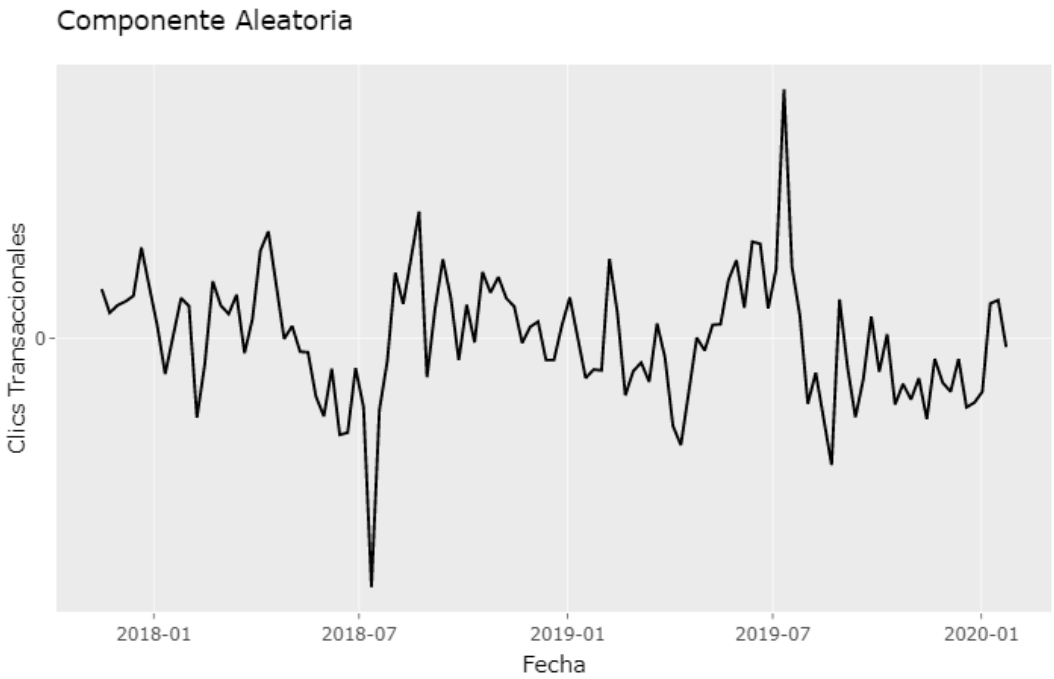

Figura 18: Serie temporal de la componente aleatoria.

A continuación, se debe analizar el comportamiento estacionario de esta serie, para ello, podemos observar el comportamiento de la ACF y PACF, y hacer los test de Box-Pierce y de Ljung-Box [15].

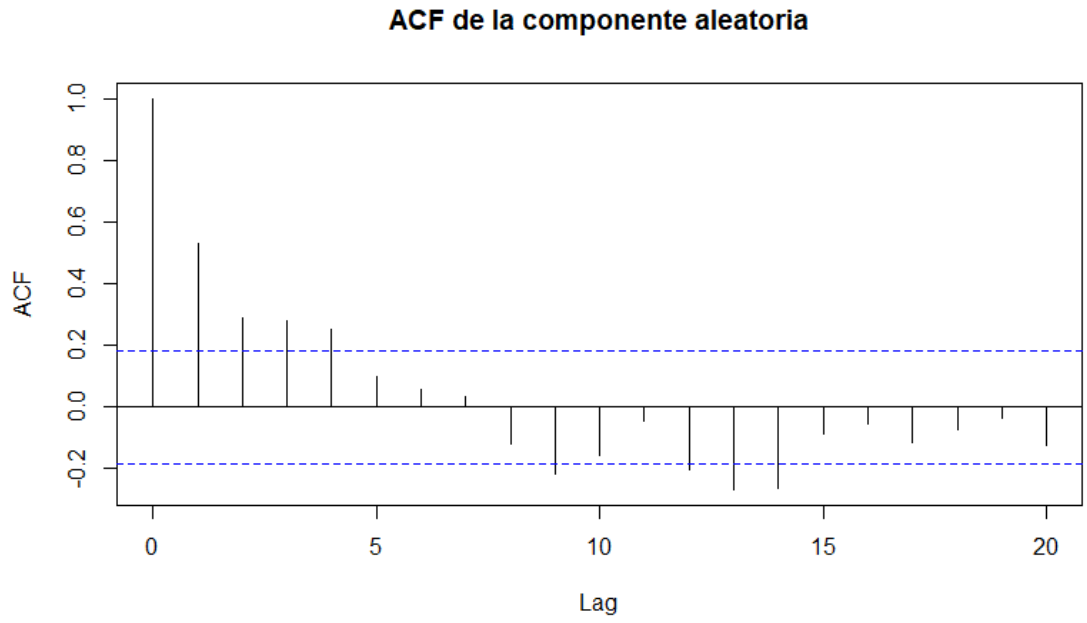

Figura 19: ACF de la componente aleatoria.

En el gráfico de la ACF (figura 19) se puede observar claramente como decrecen lentamente las barras por lo que parece que todavía queda correlación entre las variables. En el gráfico de la PACF (figura 20) existe una correlación significativa en el desfase 1 seguida de correlaciones que no son significativas. Este patrón puede indicar un término autorregresivo de orden 1. Estos gráficos son muy útiles para determinar que Modelo ARMA ajustar. Parece razonable probar con un AR(1).

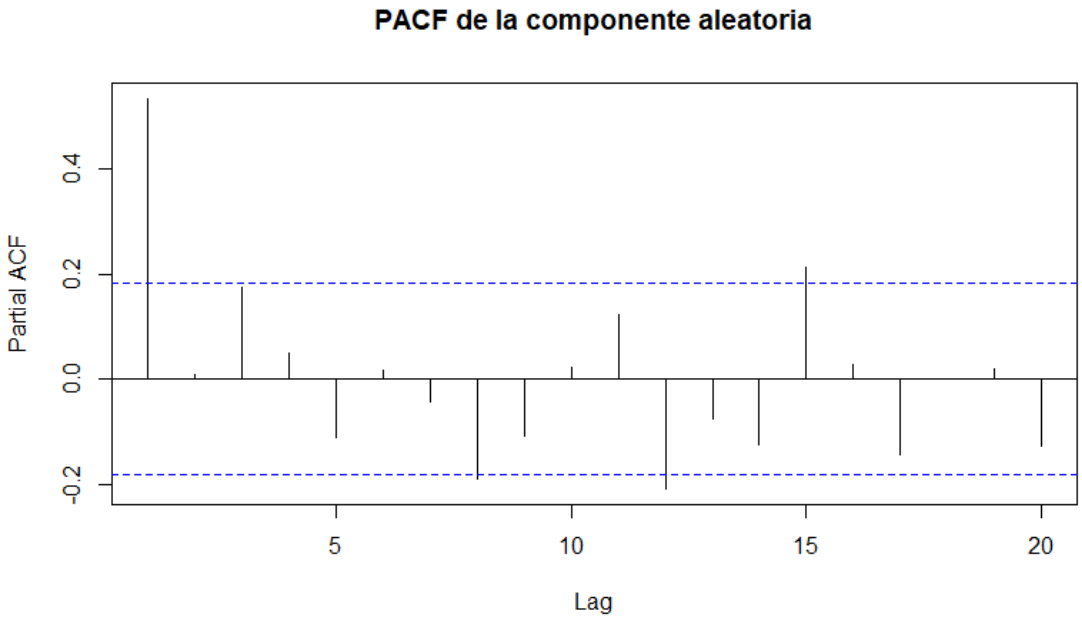

Figura 20: PACF de la componente aleatoria.

Una vez vistos los correlogramas se debe efectuar los test para verificar el comportamiento de las autocorrelaciones.

Primero, se efectuará la prueba de Ljung-Box, se puede definir de la siguiente manera.

H_0: Los datos se distribuyen de forma independiente (es decir, en partivular, las correlaciones en la población de la que se toma la muestra son 0, de modo que cualquier correlación observada en los datos es el resultado de la aleatoriedad del proceso de muestreo).

H_1: Los datos no se distribuyen de forma independiente.

La fórmula de la prueba es:

$$Q = n(n+2)\sum_{k=1}^{h}\frac{\hat{\rho}_k^2}{n-k} \tag{5.17}$$

donde $\hat{\rho}_k^2$ es la autocorrelación de la muestra en el retraso k, h el número de retrasos que se están probando y n el tamaño de la muestra.

Los resultados del test Ljung-Box han sido los siguientes:

Ljung-Box test

X-squared = 33.459, h = 1, p-value = 7.278e-09

Estos resultados indican que los datos no se distribuyen de forma independiente, para tener un resultado más seguro se efectuará otro test. El test de Box-Pierce utiliza la prueba estadística con las hipótesis que se indican anteriormente, la fórmula de la prueba es:

$$Q_{BP} = n \sum_{k=1}^{h} \hat{\rho}_k^2 \tag{5.18}$$

Los resultados del test Box-Pierce han sido los siguientes:

Box-Pierce test

X-squared = 32.601, h = 1, p-value = 1.132e-08

Dados estos resultados se refuta la hipótesis nula de los test y, por lo tanto, podemos afirmar que la componente aleatoria que se ha encontrado no es un ruido blanco.

Solo se estudiado para $h = 1$ porqué si se refuta la hipótesis con un retardo, lo hará en todos.

Se hará un último test de Shapiro-Wilk para comprobar si los datos siguen una distribución normal. El Test de Shapiro–Wilk se usa para contrastar la normalidad de un conjunto de datos. Se plantea como hipótesis nula que una muestra $x_1, ..., x_n$ proviene de una población normalmente distribuida.

El estadístico del test es:

$$W = \frac{(\sum_{i=1}^{n} a_i x_{(i)})^2}{\sum_{i=1}^{n} (x_i - \bar{x})^2} \tag{5.19}$$

donde $x_{(i)}$ es el número que ocupa la i-ésima posición en la muestra (con la muestra ordenada de menor a mayor), $\bar{x}$ es la media muestral y las variables a_i se calculan con los valores medios del estadístico ordenado y la matriz de covarianzas de ese estadístico de orden.

La hipótesis nula se rechazará si W es demasiado pequeño.

Los resultados del test Shapiro-Wilk han sido los siguientes:

Shapiro-Wilk normality test

W = 0.96152, p-value = 0.002215

Tenemos una W muy alta pero dado que el p-valor ha salido tan bajo, se debe recha-

zar la hipótesis nula.

Si observamos su histograma (figura 21) se observa que la componente aleatoria no sigue una distribución normal.

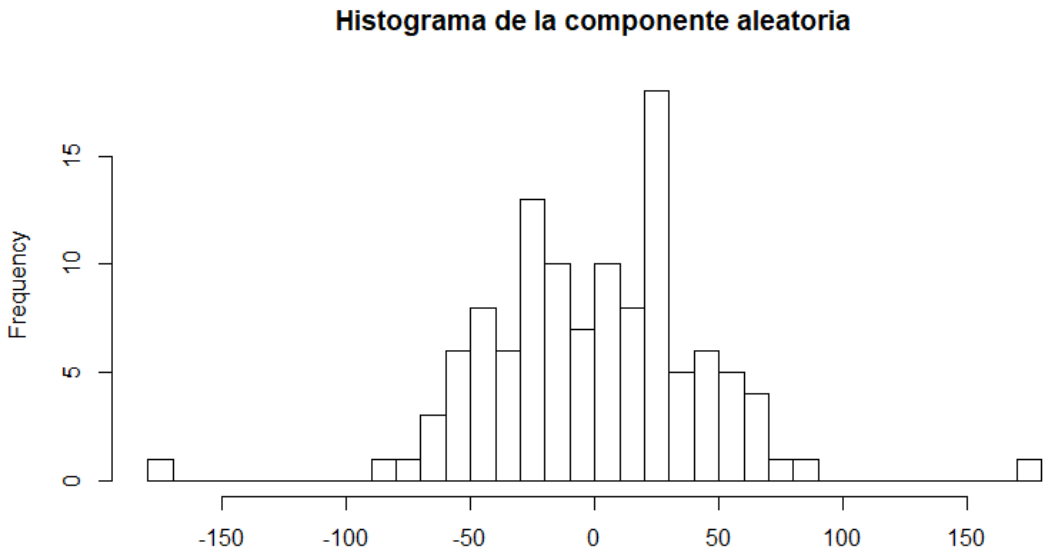

Figura 21: Histograma de la componente aleatoria.

Llegados a este punto del análisis, se debe buscar un buen ajuste ARMA para la componente aleatoria. En el análisis del PACF se ha observado que un AR(1) puede ajustarse bien a la serie temporal. Aún así, lo comprobaremos con la función *auto.arima* de la librería *forecast* de Rstudio, esta comprueba que ajuste es el mejor para los datos con los que se trabaja.

Consola de Rstudio

>auto.arima(arma, stepwise = T, approximation = F, trace= T)

ARIMA(2,0,2) with non-zero mean : 1165.554
ARIMA(0,0,0) with non-zero mean : 1197.569
ARIMA(1,0,0) with non-zero mean : 1161.414
ARIMA(0,0,1) with non-zero mean : 1167.659
ARIMA(0,0,0) with zero mean : 1195.497
ARIMA(2,0,0) with non-zero mean : 1163.554
ARIMA(1,0,1) with non-zero mean : 1163.541
ARIMA(2,0,1) with non-zero mean : 1165.499
ARIMA(1,0,0) with zero mean : 1159.306
ARIMA(2,0,0) with zero mean : 1161.409
ARIMA(1,0,1) with zero mean : 1161.396
ARIMA(0,0,1) with zero mean : 1165.55
ARIMA(2,0,1) with zero mean : 1163.313

Best model: ARIMA(1,0,0) with zero mean

Gracias a Rstudio se puede verificar lo que se ha observado en el análisis del PACF. Por esta razón se va a proceder a ajustar un modelo autoregresivo AR(1).

En la figura 22 se puede observar la comparativa entre la componente aleatoria que se ha extraído y el ajuste AR(1).

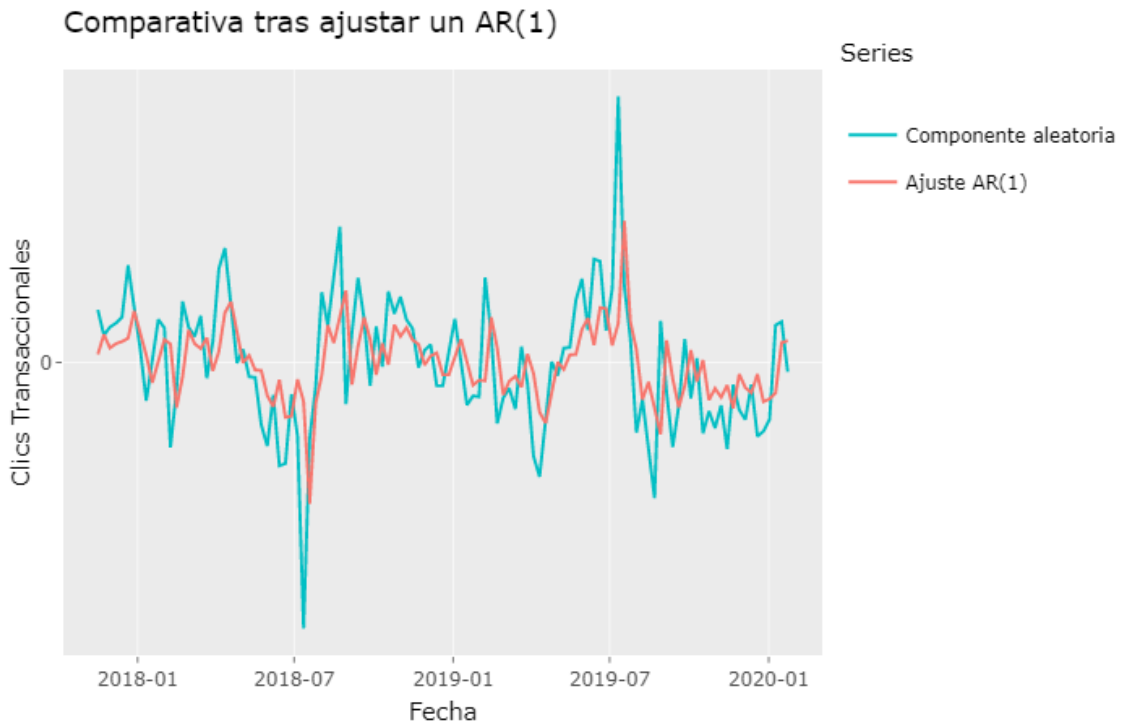

Figura 22: Comparativa de la componente aleatoria con el ajuste AR(1).

La serie del ajuste tiene los siguientes coeficientes:

Coeficientes del AR(1):

$\phi_1 = 0{,}5309$
$\delta = 0{,}0784$

σ^2 estimada : 1357
AIC=1159.2 AICc=1159.31 BIC=1164.69

Definición 5.4. *El criterio de información de Akaike (AIC) [16] es una medida de la calidad relativa de un modelo estadístico, para un conjunto dado de datos.*
En el caso general, el AIC es

$$AIC = 2k - 2ln(L) \tag{5.20}$$

donde k es el número de parámetros en el modelo estadístico , y L es el máximo valor de la función de verosimilitud para el modelo estimado.

Definición 5.5. *Cuando el tamaño de la muestra es finito se utiliza la siguiente corrección:*

$$AICc = AIC + \frac{2k^2 + 2k}{n - k - 1} \tag{5.21}$$

donde k es el número de parámetros en el modelo estadístico y n el tamaño de la muestra.

Definición 5.6. *En estadística, el criterio de información bayesiano (BIC) es un criterio para la selección de modelos entre un conjunto finito de modelos.*

$$BIC = -2 \cdot \ln L + k \ln(n) \tag{5.22}$$

donde k es el número de parámetros en el modelo estadístico , L es el máximo valor de la función de verosimilitud para el modelo estimado y n el tamaño de la muestra.

Una vez identificado el ajuste de la componente aleatoria se deben analizar sus residuos para ver como de bueno es el modelo que se quiere ajustar. Dependiendo del comportamiento de los residuos, se podrá hacer una mejor o peor predicción.

Primero se observará gráficamente el comportamiento de los residuos.

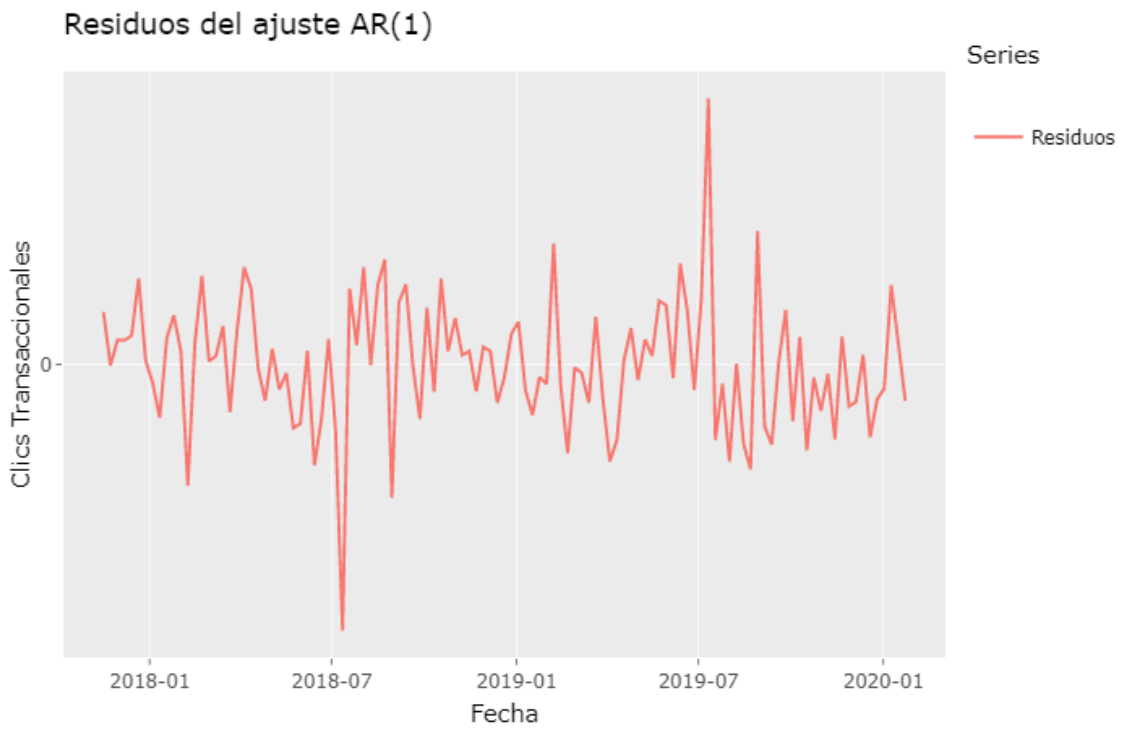

Figura 23: Residuos del ajuste AR(1).

En la figura 23 se puede observar algo que ya se parece más a un ruido blanco. Para analizarlo seguiremos la misma metodología que se ha seguido para el análisis de la componente aleatoria. Primero se analizarán los correlogramas:

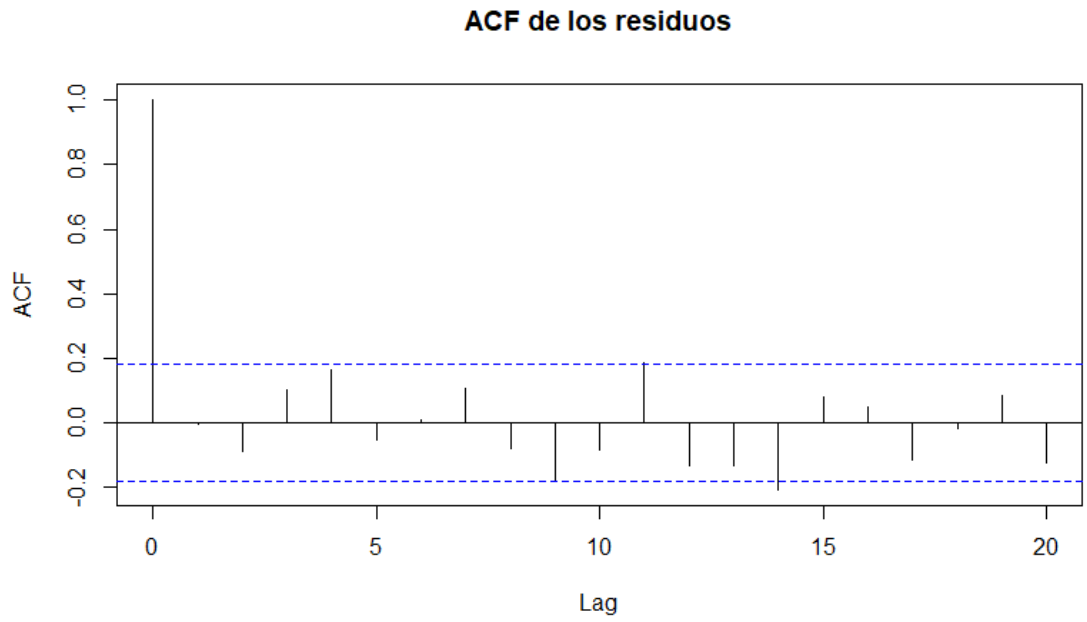

Figura 24: ACF del ajuste AR(1).

En el ACF de los residuos (figura 24) se puede observar una gran diferencia con el ACF de la componente aleatoria (figura 19). En este, se pierde el lento decrecimiento de las barras, lo cual es un buen indicador.

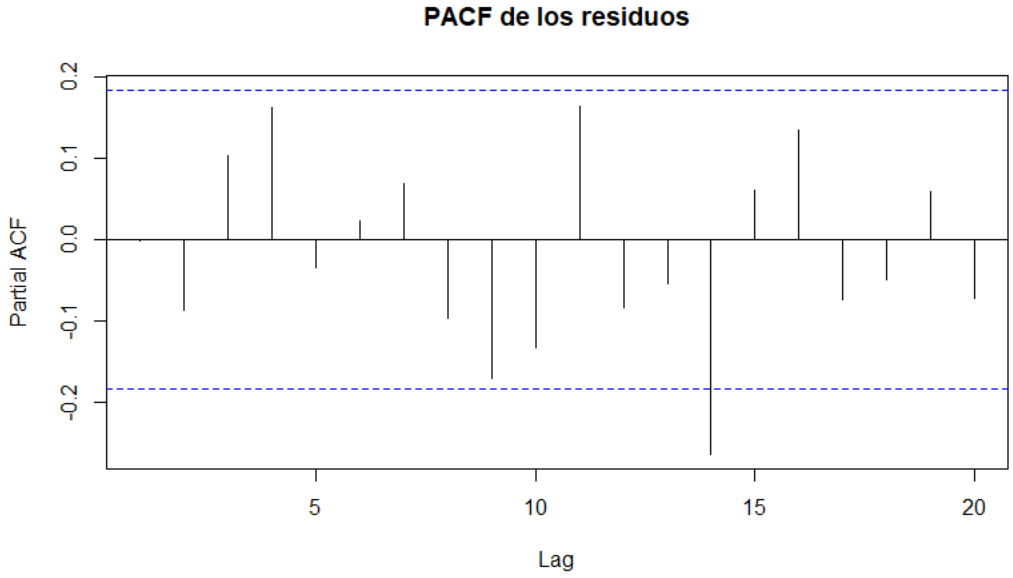

Figura 25: PACF del ajuste AR(1).

En el PACF de los residuos (figura 25) ha desaparecido la correlación significativa del desfase 1 que se había observado en la figura 20. Esto también puede indicar que los residuos son un ruido blanco.

A continuación, se debe analizar numéricamente el comportamiento de los residuos, también se deben efectuar los test realizados anteriormente para poder confirmar que la serie de los residuos es un ruido blanco.

Las principales características de los residuos son:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-149.20443	-21.24918	0.37083	-0.07447	19.74067	149.20443

$\sigma = 36{,}84$

Tienen media 0, una desviación estándar de 36,84 y parece que las variables aleatorias están distribuidas de la misma forma.

Los test realizados a los residuos son:

Ljung-Box test

X-squared = 0.00068, h = 1, p-value = 0.979

Box-Pierce test

X-squared = 0.00066, h = 1, p-value = 0.979

Ahora, los resultados de los test dan unos p-valores muy altos por lo que se acepta la hipótesis nula y se puede afirmar que los datos se distribuyen de forma independiente.

Se han comprobado los resultados del test de Ljung-Box para diferentes retardos y se sigue aceptando la hipótesis nula:

X-squared = 0.8862, h = 2, p-value = 0.642
X-squared = 2.1368, h = 3, p-value = 0.545
X-squared = 5.4832, h = 4, p-value = 0.241

Con estos resultados, se puede decir que los residuos del ajuste AR(1) son un ruido IID y, por lo tanto, la predicción que se haga tiene que ser correcta.

Por último, se debe ver si los residuos siguen una distribución normal tal y como se ha hecho con la componente aleatoria.

Shapiro-Wilk normality test

W = 0.96219, p-value = 0.002505

El test de Shapiro-Wilk nos dice que los residuos siguen sin tener una distribución normal. Por lo que se hace un histograma para entender mejor como se distribuye.

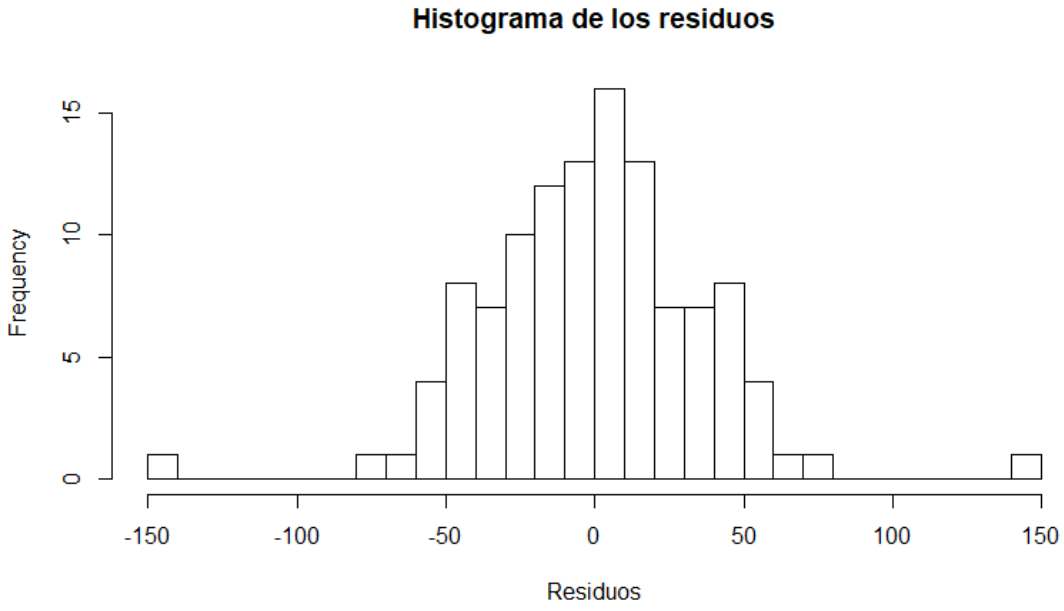

Figura 26: Histograma de los residuos.

Para acabar con el análisis de los residuos, se intenta estimar la ley de los residuos con la librería de Rstudio *univariateML* y nos da como resultado una distribución Weibull con parámetro de forma $k = 4{,}3$ y parámetro de escala $\lambda = 168{,}38$ [17].

Se le hacen algunos gráficos de diagnóstico (figura 27) y parece ser una buena estimación para la ley de los residuos.

Definición 5.7. *Q-Q Plot es un método gráfico para el diagnóstico de diferencias entre la distribución de probabilidad de una población de la que se ha extraído una muestra aleatoria y una distribución usada para la comparación.*

Definición 5.8. *P-P Plot es un gráfico de probabilidad para evaluar qué tan de cerca están de acuerdo dos conjuntos de datos, que traza las dos funciones de distribución acumulativa entre sí.*

Definición 5.9. *Un gráfico de función de distribución acumulativa (CDF) muestra la función de distribución acumulativa empírica de los datos.*

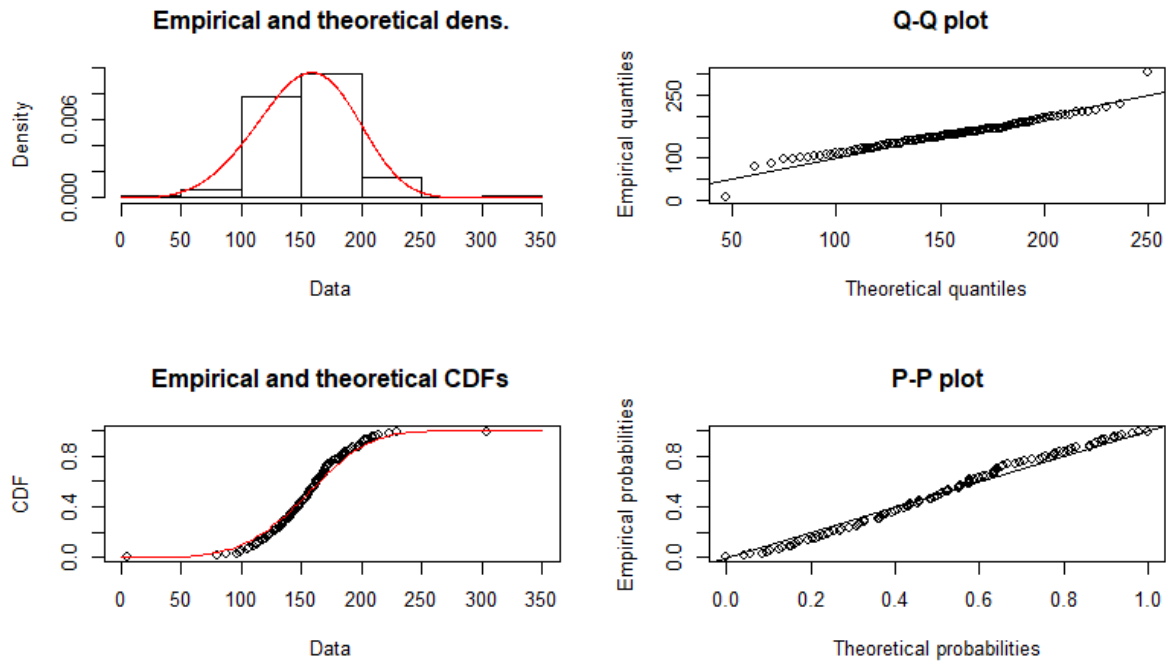

Figura 27: Gráficos de diagnóstico de la estimación.

Después de hacer un análisis exhaustivo de los residuos y observar que son un ruido IID se va a proceder a hacer la predicción.

Para este proceso se va a utilizar la librería *forecast*[18] de Rstudio.

Primero se ha hecho la predicción de 30 semanas de la componente aleatoria que se ha estudiado anteriormente y se han obtenido los siguientes resultados:

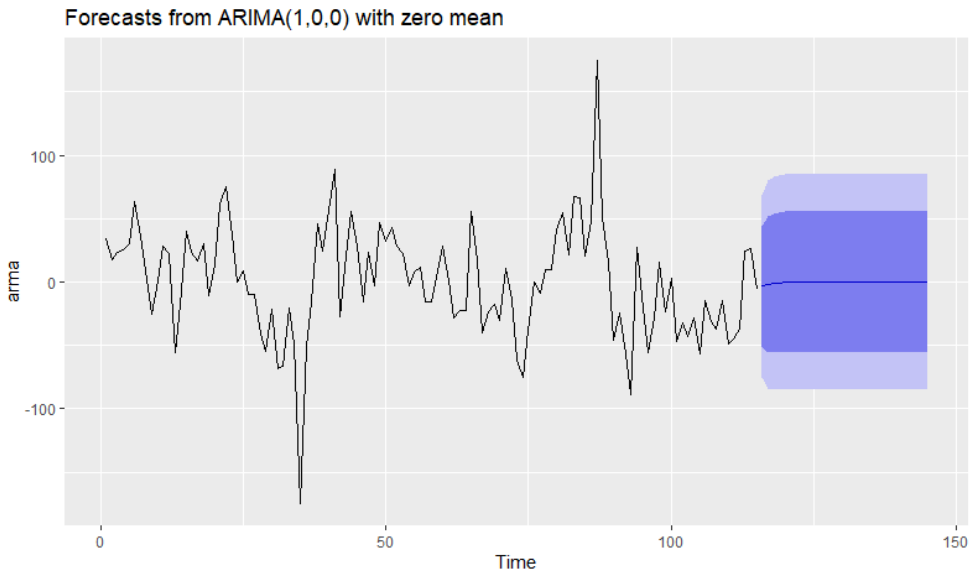

Figura 28: Predicción de AR(1) de la componente aleatoria.

Como se puede observar en la figura 28, sale un resultado bastante estable, esto puede ser debido al ajuste dado que el modelo AR(1) solo evalúa el valor del instante anterior.

Las franjas azules definen los intervalos de confianza, el oscuro con un 80 % y el claro con un 95 %.

Llegados a este punto, solo falta incorporar la tendencia y la estacionalidad para así obtener la predicción de los clics transaccionales (figura 29).

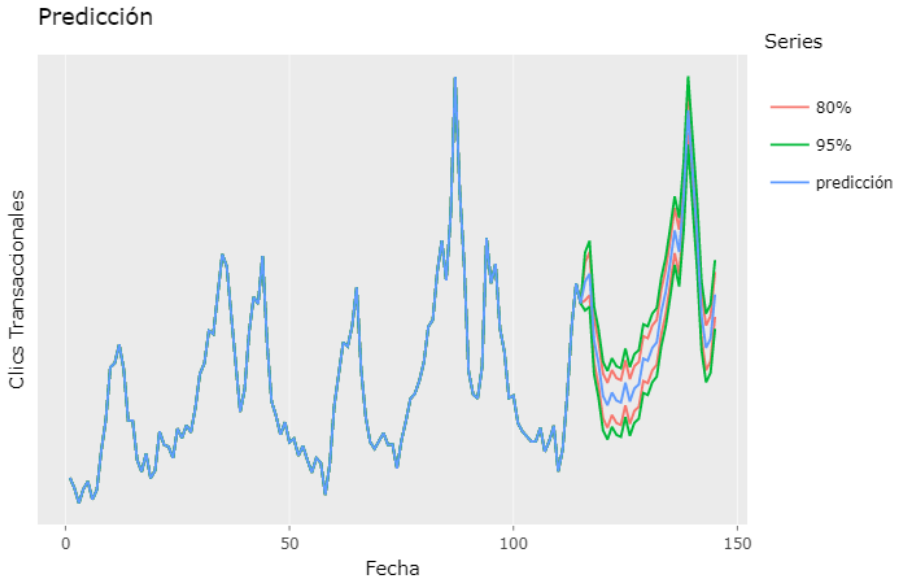

Figura 29: Predicción AR(1) de los clics transaccionales.

Para poder observar mejor los intervalos de confianza, se ha hecho un zoom en la figura 29. En este caso, las lineas rojas delimitan el intervalo de confianza del 80 % y las lineas verdes delimitan el del 95 %.

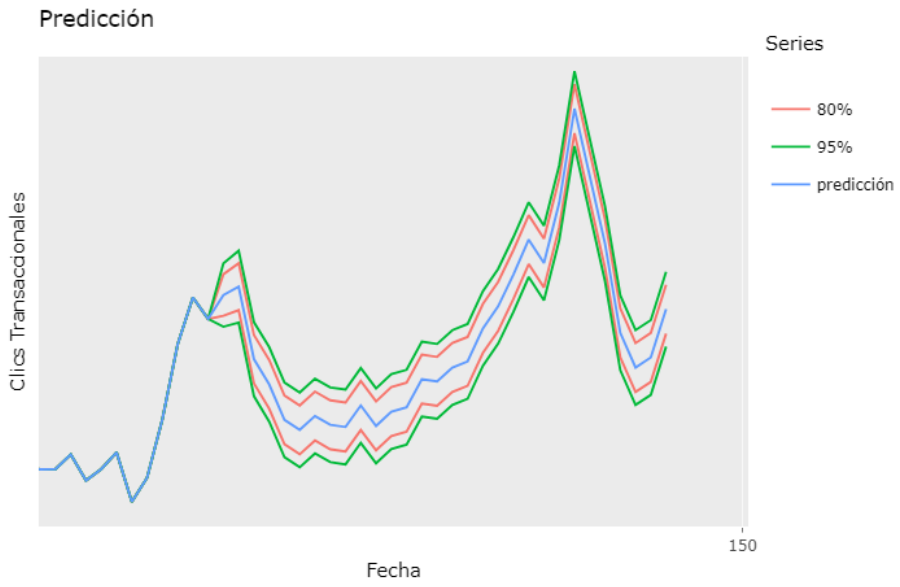

Figura 30: Zoom a la predicción AR(1) de los clics transaccionales.

Para finalizar con el apartado de análisis, se va a hacer una predicción con la librería *forecast* de Rstudio sin hacer la descomposición de la serie temporal. Esta librería, tiene una función a la que si se le introduce una serie temporal, la función la analiza y hace una predicción. En la figura 31, se puede observar qué devuelve la función si se le da la serie temporal de clics transaccionales.

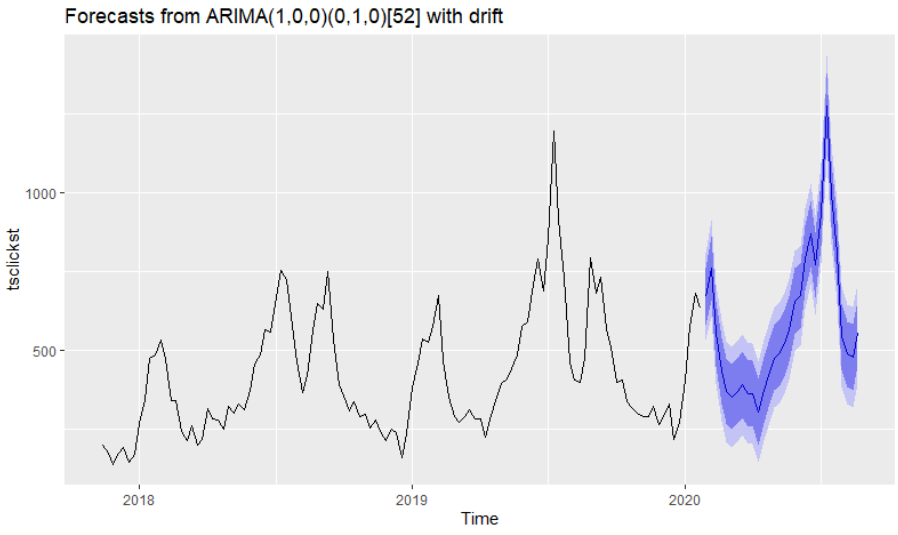

Figura 31: Predicción realizada con Rstudio de los clics transaccionales.

Como se puede observar en la figura 31, la predicción con *forecast* es prácticamente la misma. Al no haber hecho descomposición, *forecast* utiliza un modelo que tiene en cuenta la estacionaridad y la estacionalidad.

6. Resultados

Al principio de este trabajo se han planteado diferentes objetivos. En este apartado se comentará y se discutirá acerca de los resultados hallados.

Con la información de Google se ha podido observar diferentes patrones de búsqueda según la intencionalidad. Se ha podido clasificar la intención de búsqueda del usuario según si estaba buscando información, si buscaba información sobre algo concreto o si era una búsqueda más cercana a una transacción. Gracias a esta diferenciación por categoría, tenemos la posibilidad de hacer un análisis de clic, impresiones o posicionamiento pero diferenciado por la intencionalidad de la búsqueda. Gracias a esto, también se ha podido verificar que la categoría transaccional es la que tiene más relación con los leads.

Como se ha comentado anteriormente, durante la transformación de los datos se han creado múltiples variables muy valiosas. Gracias a una matriz de correlaciones, se ha podido encontrar y analizar unas correlaciones muy altas en alguna de estas variables. Por ejemplo, la correlación entre los clics transaccionales y los leads es de un 0,88.

La creación de tantas variables no solo a servido para la matriz de correlación. Aunque la mayoría de esta información no se haya visto en el análisis, estas variables pueden ser buenos KPI's para posibles dashboards.

A continuación se pueden ver dos posibles ejemplos de dashboard interactivos con datos simulados:

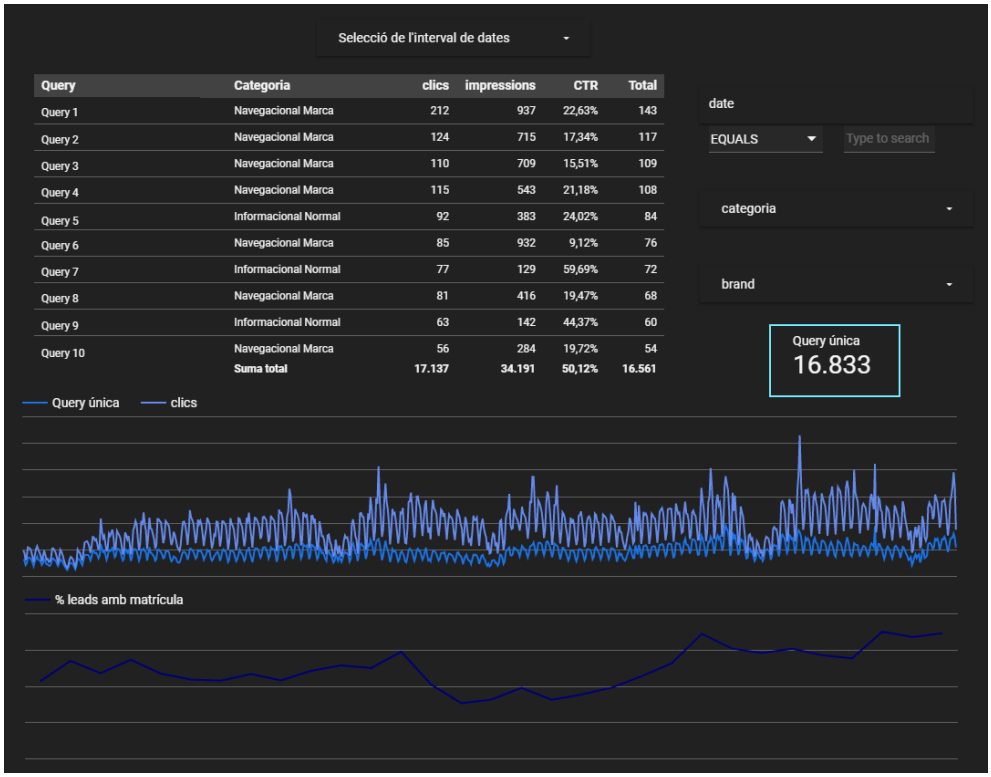

Figura 32: Dashboard de análisis por fecha.

En la figura 32, se puede observar un panel que busca facilitar el análisis por fecha. En la parte derecha superior se puede seleccionar una fecha concreta o un intervalo de tiempo, de manera que, el panel te muestra diferente información: Las diez querys más buscadas con toda su información de categoría, clics, impresiones, CTR o recuento; Después un gráfico con la comparativa de dos series temporales durante el intervalo de tiempo seleccionado y, por último, otro gráfico con una variable que recoge el porcentaje de matrícula por lead que se hace cada día.

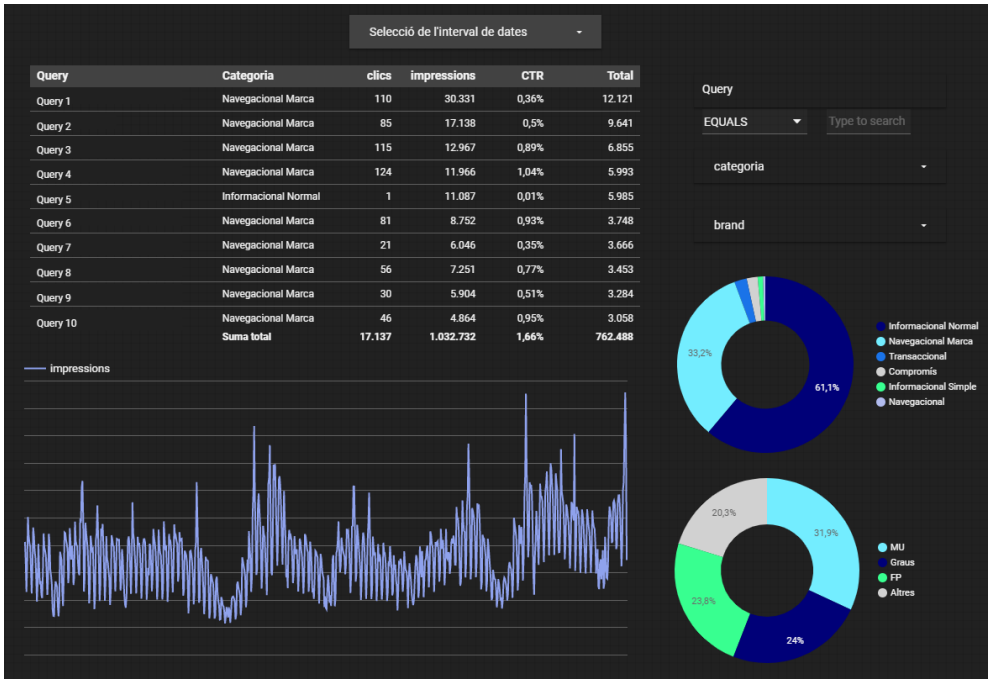

Figura 33: Dashboard de análisis por query.

En la figura 33, se puede observar un panel diferente que busca facilitar el análisis por query. En la parte derecha superior hay un buscador por palabra, para poder ver la lista de querys, con toda su información como en el dashboard anterior, que contienen esa palabra. También hay una serie temporal de impresiones para las palabras seleccionadas y, debajo del buscador, dos gráficos de porciones con porcentajes. El primero indica las categorías de las querys y el segundo indica la tipología de estudios del enlace que ha aparecido en Google tras buscar las querys de la lista.

Ambos dashboard se han creado con el sowtware para visualización de datos de Google, Data Studio.

Se han creado series temporales de algunas de las variables. Estas series son muy interesantes porqué permiten ver el comportamiento que tienen algunas variables durante el año. Durante el estudio se ha podido observar como la variable de clics transaccionales está muy correlacionada con importantes variables como son las de leads, leads con matrícula y matrículas. Es por eso que se ha decidido hacer un buen análisis de la serie temporal generada por la variable de clics transaccionales. En este análisis, se ha podido observar tanto su comportamiento respecto al tiempo, como sus máximos y mínimos. Además, se ha encontrado una tendencia positiva y una parte estacional anual, que gra-

cias a la descomposición de las series temporales, ha permitido encontrar la componente aleatoria.

Por último, se ha estudiado el comportamiento de la componente aleatoria, se ha observado que era poco estacionario y que se podía ajustar bien a un modelo AR(1).

Después de ajustar el modelo autorregresivo, se han estudiado sus residuos. Tras realizar los test Box-Ljung y Box-Pierce, se puede afirmar que las variables se distribuyen de forma independiente, por lo tanto, los residuos se comportan como un ruido IDD.

Dado que el test de Shapiro-Wilk ha confirmado que los residuos no siguen una distribución normal, se ha hecho una estimación de la ley de los residuos y se ha encontrado que una distribución Weibull la estima bien.

Como se ha visto que los residuos se comportan como un ruido blanco, se ha procedido a hacer una predicción de la componente aleatoria. A la cual, se le ha incorporado la componente estacional y la tendencia para obtener una predicción de los clics transaccionales. La predicción indica que habrá primero una caída y, posteriormente, un pico con máximo absoluto.

Para finalizar con la predicción, se prueba a calcular la predicción pero, esta vez, con una herramienta de *forecast* que hace todo el análisis automáticamente. Los resultados de la predicción de *forecast* son muy parecidos a los de la predicción clásica, por lo tanto, se puede afirmar que para esta serie temporal esta herramienta trabaja muy bien.

7. Conclusiones

Hoy en día, la gran mayoría de las empresas que se precien tienen un proceso de marketing integrado o basado en datos estratégicos. Son muchos los campos en los que se desea conocer el comportamiento futuro de ciertos fenómenos con el objetivo de adelantarse a los acontecimientos. Debido a esta necesidad aparecen estudios como este donde se busca optimizar las herramientas de marketing y también estudios de series temporales cuya principal finalidad es predecir lo que ocurrirá con una variable en el futuro a partir del comportamiento de esa variable en el pasado y de otros factores que puedan influir.

Por un lado, en este estudio, se ha podido observar que los datos con los que se trabaja normalmente en marketing son sólo la punta del iceberg de toda la información posible. Esto, no se debe malinterpretar. No siempre tener más información es mejor, a veces, demasiada información puede crear confusión. Es importante saber filtrar y seleccionar los datos que aportan un valor añadido. Por ejemplo, cuando se ha estudiado las correlaciones entre las series temporales se han encontrado índices mejores que los de los clics transaccionales, pero la información que podrán aportar estas series no tiene tanto valor.

También, es necesario destacar que en estudios como este, que se usan datos de Google, la muestra con la que se trabaja (más de 21 millones de entradas) es de un tamaño muy superior al normal y, por ese motivo, es de vital importancia la optimización de los códigos y el uso de hardware con gran potencial.

Como se ha comentado, se ha centrado el análisis a la serie temporal de clics transaccionales. El principal motivo ha sido los buenos índices de correlación obtenidos en la matriz de correlación, esta, ha indicado una gran relación con variables como leads, leads con matrícula y matrícula, además, gráficamente tienen un gran parecido.

Llegados a este punto, y con lo comentado anteriormente, ¿Qué valor añadido aporta esta nueva variable? Esta variable nos crea una relación entre los datos más de negocio con los datos utilizados en marketing digital. Este hecho desde marketing puede ser muy útil dado que se podrá saber qué búsquedas están más relacionadas con la transacción y, con esta información, se pueden realizar acciones como, por ejemplo, mejorar el posicionamiento de la web en esas búsquedas.

En cuanto al análisis, en la serie temporal de clics transaccionales se ha visto una tendencia alcista y una estacionalidad anual, lo más seguro es que las matrículas tengan el mismo comportamiento. También se ha logrado ajustar un modelo AR(1) al componente aleatorio y hacer una predicción de 30 semanas. El principal hecho que puede mejorar la predicción es seguir recogiendo los datos, cuanto más grande es la muestra mejores son los resultados.

Por último, es importante hacer una reflexión sobre el final del apartado de predicción. Se ha querido comparar el proceso clásico de predicción con el proceso automático que permiten algunas librerías de Rstudio. Tras la comparativa, se puede concluir que los resultados son muy parecidos y, por lo tanto, estas herramientas tienen un gran potencial.

8. Anexo

Daydata.R

```
library (bigrquery)
library(stringr)
library(dplyr)
library(readxl)
library(xlsx)
library(readxl)

#agregamos por data el sumatorio de clicks y impressiones, y la media de posicion.

xx <- aggregate(kwdata$clicks, list(date = kwdata$date), sum)
xx$impressions <- aggregate(kwdata$impressions, list(date = kwdata$date), sum)[2]
#Se utiliza el agregado de posici n que estipula Google
kwdata$posaux <- kwdata$position * kwdata$impressions
xx$positionaux <- round(aggregate(kwdata$posaux, list(date = kwdata$date), sum)[2],4)
xx$position <- xx$positionaux/xx$impressions
xx$impressions <- as.numeric(unlist(xx$impressions))
xx$position <- as.numeric(unlist(xx$position))

#agregamos recuento, ctr, cctr y rctr.

c <- data.frame(table(kwdata$date))
xx$recuento <- c$Freq
xx$ctr <- xx$x/xx$impressions
xx$cctr <- xx$x*xx$ctr
xx$rctr <- xx$recuento*xx$ctr

#Creamos una columna para cada categoria, en cada kw le pongo un 1
#en la columna de su categoria y un 0 en las otras.

yy <- data.frame(kwdata$date, kwdata$clicks, kwdata$impressions, kwdata$categoria)
yy$IN <- ifelse(grepl(pattern = 'IN', x = yy$kwdata.categoria)==TRUE,'1','0')
yy$IS <- ifelse(grepl(pattern = 'IS', x = yy$kwdata.categoria)==TRUE,'1','0')
yy$NM <- ifelse(grepl(pattern = 'NM', x = yy$kwdata.categoria)==TRUE,'1','0')
yy$N <- ifelse(grepl(pattern = '\\bN\\b', x = yy$kwdata.categoria)==TRUE,'1','0')
yy$T <- ifelse(grepl(pattern = 'T', x = yy$kwdata.categoria)==TRUE,'1','0')
yy$C <- ifelse(grepl(pattern = 'C', x = yy$kwdata.categoria)==TRUE,'1','0')
yy$IN <- as.numeric(yy$IN)
yy$IS <- as.numeric(yy$IS)
yy$NM <- as.numeric(yy$NM)
yy$N <- as.numeric(yy$N)
yy$T <- as.numeric(yy$T)
yy$C <- as.numeric(yy$C)

#Tambien lo hacemos por clicks por categoria y por impresi n

yy$clickIN <- yy$IN*yy$kwdata.clicks
yy$clickIS <- yy$IS*yy$kwdata.clicks
yy$clickNM <- yy$NM*yy$kwdata.clicks
yy$clickN <- yy$N*yy$kwdata.clicks
yy$clickT <- yy$T*yy$kwdata.clicks
yy$clickC <- yy$C*yy$kwdata.clicks

yy$impIN <- yy$IN*yy$kwdata.impressions
yy$impIS <- yy$IS*yy$kwdata.impressions
yy$impNM <- yy$NM*yy$kwdata.impressions
yy$impN <- yy$N*yy$kwdata.impressions
yy$impT <- yy$T*yy$kwdata.impressions
yy$impC <- yy$C*yy$kwdata.impressions

#Hacemos los sumatorios de las columnas de las categorias,
#clics categorias, impresion categoria y lo agregamos por data.

xx$IN <- round(aggregate(yy$IN, list(date = yy$kwdata.date), sum)[2],10)
xx$IS<- round(aggregate(yy$IS, list(date = yy$kwdata.date), sum)[2],11)
```

```
xx$NM <- round(aggregate(yy$NM, list(date = yy$kwdata.date), sum)[2],12)
xx$N <- round(aggregate(yy$N, list(date = yy$kwdata.date), sum)[2],13)
xx$T <- round(aggregate(yy$T, list(date = yy$kwdata.date), sum)[2],14)
xx$C <- round(aggregate(yy$C, list(date = yy$kwdata.date), sum)[2],15)

xx$clickIN <- round(aggregate(yy$clickIN, list(date = yy$kwdata.date), sum)[2])
xx$clickIS<- round(aggregate(yy$clickIS, list(date = yy$kwdata.date), sum)[2])
xx$clickNM <- round(aggregate(yy$clickNM, list(date = yy$kwdata.date), sum)[2])
xx$clickN <- round(aggregate(yy$clickN, list(date = yy$kwdata.date), sum)[2])
xx$clickT <- round(aggregate(yy$clickT, list(date = yy$kwdata.date), sum)[2])
xx$clickC <- round(aggregate(yy$clickC, list(date = yy$kwdata.date), sum)[2])

xx$impIN <- round(aggregate(yy$impIN, list(date = yy$kwdata.date), sum)[2])
xx$impIS<- round(aggregate(yy$impIS, list(date = yy$kwdata.date), sum)[2])
xx$impNM <- round(aggregate(yy$impNM, list(date = yy$kwdata.date), sum)[2])
xx$impN <- round(aggregate(yy$impN, list(date = yy$kwdata.date), sum)[2])
xx$impT <- round(aggregate(yy$impT, list(date = yy$kwdata.date), sum)[2])
xx$impC <- round(aggregate(yy$impC, list(date = yy$kwdata.date), sum)[2])

#Anadimos sesiones y matriculas
 #S'ha de treure el 17-1-2020 perque no es van recollir dades
sessio_per_data <- sessio_per_data[-793,]

xx$sessio_orgxclic <- sessio_per_data$sessio_org/xx$x
xx$sessio_orgximpressio <- sessio_per_data$sessio_org/xx$impressions
xx$sctr <- sessio_per_data$sessio_org*xx$ctr

dadesgraumu$Data <- as.Date(dadesgraumu$Data)
matriculas<-subset(dadesgraumu, Data >= as.Date("2017-11-16")
                   & Data <= as.Date("2020-02-01"), select = c(Grau, MU))
matriculas$total <- matriculas$Grau+matriculas$MU

#S'ha de treure el 17-1-2020 perque no es van recollir dades
matriculas <- matriculas[-793,]
#Anadimos leads , leads que acaban en matricula y dias en matricularse desde el lead

#load("~/dadesleadmat.Rdata")

#Se limpia y ordena para subir a Big Query.

daydata <- data.frame(xx$date, xx$x, xx$impressions, xx$recuento,
                      xx$position, xx$ctr, xx$cctr, xx$rctr, xx$sctr,
                      matriculas$total,
                    matriculas$Grau, matriculas$MU, sessio_per_data$sessio_org,
                    sessio_per_data$sessio, xx$sessio_orgxclic,
                    xx$sessio_orgximpressio, dadesleadmat$leads,
                    dadesleadmat$leadsmat, dadesleadmat$mitj_dies_mat,
                    dadesleadmat$leads_grau,
                    dadesleadmat$leads_mu, dadesleadmat$leadsmat_grau,
                    dadesleadmat$leadsmat_mu, dadesleadmat$mitj_dies_mat_grau,
                    dadesleadmat$mitj_dies_mat_mu, dadesleadmat$leads_org,
                    dadesleadmat$leadsmat_org, dadesleadmat$mitj_dies_mat_org,
                    dadesleadmat$leads_grau_org, dadesleadmat$leads_mu_org,
                    dadesleadmat$leadsmat_grau_org, dadesleadmat$leadsmat_mu_org,
                    dadesleadmat$mitj_dies_mat_grau_org, dadesleadmat$mitj_dies_mat_mu_org,
                    xx$IS, xx$IN, xx$N, xx$NM, xx$C, xx$T, xx$clickIS, xx$clickIN, xx$clickN,
                    xx$clickNM, xx$clickC, xx$clickT, xx$impIS, xx$impIN,
                    xx$impN, xx$impNM, xx$impC, xx$impT)


names(daydata) <- c("date", "clics", "impressions", "recuento", "posicio", "CTR",
                    "cctr", "rctr", "sctr", "mat", "mat_grau", "mat_mu",
                    "sessio_org", "sessio", "sessio_orgxclic", "sessio_orgximpressio",
                    "leads", "leadsmat", "mitj_dies_mat", "leads_grau",
                    "leads_mu", "leadsmat_grau", "leadsmat_mu", "mitj_dies_mat_grau",
                    "mitj_dies_mat_mu", "leads_org", "leadsmat_org",
                    "mitj_dies_mat_org", "leads_grau_org", "leads_mu_org", "leadsmat_grau_org",
                    "leadsmat_mu_org", "mitj_dies_mat_grau_org", "mitj_dies_mat_mu_org", "IS",
                    "IN", "N", "NM", "C", "T", "clicksIS", "clicksIN", "clicksN", "clicksNM",
                    "clicksC", "clicksT", "impIS", "impIN", "impN", "impNM", "impC", "impT")
```

Weekdata.R

```
library(ggplot2)
library(plotly)
library (bigrquery)
library(stringr)
library(dplyr)
library(readxl)
library(normalr)
library(rJava)
library(xlsx)
library(readxl)

#Poner la primera fecha que tienes

datainici = as.Date("2017-11-16")

#Poner la BBDD por fecha

datos <- daydata

#Algoritmo que divide las filas entre 7 para hacer la
#transformacion a semana.
#Si el numero no es multiple de de 7
#se quitan dias del final para que no de error.

n <- nrow(datos)
aux <- n%%7
for (i in 0:(aux-1)) {
  datos <- datos[-(n-i),]
}
datos$set <- rep(seq(1, n%/%7), each = 7)

#Agregamos los datos que teniamos por fecha a por semana.

fichero <- aggregate(datos$clics, by=list(datos$set), FUN=sum)
fichero$impressions <- aggregate(datos$impressions, by=list(datos$set), FUN=sum)[2]
fichero$recuento <- aggregate(datos$recuento, by=list(datos$set), FUN=sum)[2]

#fichero$posicion <- aggregate(datos$posicio, by=list(datos$set), FUN=mean)[2]

#metodo de google para agragar posicion:
datos$posaux <- datos$posicio * datos$impressions
fichero$positionaux <- aggregate(datos$posaux, by=list(datos$set), sum)[2]
fichero$posicio <- fichero$positionaux/fichero$impressions

fichero$ctr <- aggregate(datos$CTR, by=list(datos$set), FUN=mean)[2]
fichero$mat <- aggregate(datos$mat, by=list(datos$set), FUN=sum)[2]
fichero$mat_grau <- aggregate(datos$mat_grau, by=list(datos$set), FUN=sum)[2]
fichero$mat_mu <- aggregate(datos$mat_mu, by=list(datos$set), FUN=sum)[2]
fichero$sessio <- aggregate(datos$sessio, by=list(datos$set), FUN=sum)[2]
fichero$sessio_org <- aggregate(datos$sessio_org, by=list(datos$set), FUN=sum)[2]
fichero$IS <- aggregate(datos$IS, by=list(datos$set), FUN=sum)[2]
fichero$IN <- aggregate(datos$IN, by=list(datos$set), FUN=sum)[2]
fichero$N <- aggregate(datos$N, by=list(datos$set), FUN=sum)[2]
fichero$NM <- aggregate(datos$NM, by=list(datos$set), FUN=sum)[2]
fichero$C <- aggregate(datos$C, by=list(datos$set), FUN=sum)[2]
fichero$T <- aggregate(datos$T, by=list(datos$set), FUN=sum)[2]
fichero$clicksIS <- aggregate(datos$clicksIS, by=list(datos$set), FUN=sum)[2]
fichero$clicksIN <- aggregate(datos$clicksIN, by=list(datos$set), FUN=sum)[2]
fichero$clicksN <- aggregate(datos$clicksN, by=list(datos$set), FUN=sum)[2]
fichero$clicksNM <- aggregate(datos$clicksNM, by=list(datos$set), FUN=sum)[2]
fichero$clicksC <- aggregate(datos$clicksC, by=list(datos$set), FUN=sum)[2]
fichero$clicksT <- aggregate(datos$clicksT, by=list(datos$set), FUN=sum)[2]
fichero$impIS <- aggregate(datos$impIS, by=list(datos$set), FUN=sum)[2]
fichero$impIN <- aggregate(datos$impIN, by=list(datos$set), FUN=sum)[2]
fichero$impN <- aggregate(datos$impN, by=list(datos$set), FUN=sum)[2]
fichero$impNM <- aggregate(datos$impNM, by=list(datos$set), FUN=sum)[2]
```

```
fichero$impC <- aggregate(datos$impC, by=list(datos$set), FUN=sum)[2]
fichero$impT <- aggregate(datos$impT, by=list(datos$set), FUN=sum)[2]

fichero$leads <- aggregate(datos$leads, by=list(datos$set), FUN=sum)[2]
fichero$leadsmat <- aggregate(datos$leadsmat, by=list(datos$set),
                              FUN=sum)[2]
fichero$mitj_dies_mat <- aggregate(datos$mitj_dies_mat, by=list(datos$set),
                                   FUN=mean)[2]
fichero$leads_grau <- aggregate(datos$leads_grau, by=list(datos$set),
                                FUN=sum)[2]
fichero$leads_mu <- aggregate(datos$leads_mu, by=list(datos$set),
                              FUN=sum)[2]
fichero$leadsmat_grau <- aggregate(datos$leadsmat_grau, by=list(datos$set),
                                   FUN=sum)[2]
fichero$leadsmat_mu <- aggregate(datos$leadsmat_mu, by=list(datos$set),
                                 FUN=sum)[2]
fichero$mitj_dies_mat_grau <- aggregate(datos$mitj_dies_mat_grau, by=list(datos$set),
                                        FUN=mean)[2]
fichero$mitj_dies_mat_mu <- aggregate(datos$mitj_dies_mat_mu, by=list(datos$set),
                                      FUN=mean)[2]

fichero$leads_org <- aggregate(datos$leads_org, by=list(datos$set),
                               FUN=sum)[2]
fichero$leadsmat_org <- aggregate(datos$leadsmat_org, by=list(datos$set),
                                  FUN=sum)[2]
fichero$mitj_dies_mat_org <- aggregate(datos$mitj_dies_mat_org, by=list(datos$set),
                                       FUN=mean)[2]
fichero$leads_grau_org <- aggregate(datos$leads_grau_org, by=list(datos$set),
                                    FUN=sum)[2]
fichero$leads_mu_org <- aggregate(datos$leads_mu_org, by=list(datos$set),
                                  FUN=sum)[2]
fichero$leadsmat_grau_org <- aggregate(datos$leadsmat_grau_org, by=list(datos$set),
                                       FUN=sum)[2]
fichero$leadsmat_mu_org <- aggregate(datos$leadsmat_mu_org, by=list(datos$set),
                                     FUN=sum)[2]
fichero$mitj_dies_mat_grau_org <- aggregate(datos$mitj_dies_mat_grau_org,
                                            by=list(datos$set), FUN=mean)[2]
fichero$mitj_dies_mat_mu_org <- aggregate(datos$mitj_dies_mat_mu_org,
                                          by=list(datos$set), FUN=mean)[2]
#Se crea sequencia de semanas para tener eje temporal.

fichero$Group.1 <- seq(datainici, datainici+7*(n%/%6)-7, by=7)

#Se limpia y ordena para subir a Big Query.

weekdata <- data.frame(fichero$Group.1, fichero$x, fichero$impressions,
                       fichero$recuento, fichero$posicion, fichero$ctr,
                       fichero$mat, fichero$mat_grau, fichero$mat_mu,
                       fichero$sessio_org, fichero$sessio,
                       fichero$leads, fichero$leadsmat, fichero$mitj_dies_mat,
                       fichero$leads_grau, fichero$leads_mu, fichero$leadsmat_grau,
                       fichero$leadsmat_mu, fichero$mitj_dies_mat_grau,
                       fichero$mitj_dies_mat_mu, fichero$leads_org, fichero$leadsmat_org,
                       fichero$mitj_dies_mat_org, fichero$leads_grau_org,
                       fichero$leads_mu_org, fichero$leadsmat_grau_org,
                       fichero$leadsmat_mu_org, fichero$mitj_dies_mat_grau_org,
                       fichero$mitj_dies_mat_mu_org, fichero$IS, fichero$IN, fichero$N,
                       fichero$NM, fichero$C, fichero$T,
                       fichero$clicksIS, fichero$clicksIN, fichero$clicksN,
                       fichero$clicksNM, fichero$clicksC, fichero$clicksT,
                       fichero$impIS, fichero$impIN, fichero$impN, fichero$impNM,
                       fichero$impC, fichero$impT)

names(weekdata) <- c("date", "clics", "impressions", "recuento", "posicio", "CTR",
                     "mat", "mat_grau", "mat_mu", "sessio_org", "sessio", "leads",
                     "leadsmat", "mitj_dies_mat", "leads_grau", "leads_mu", "leadsmat_grau",
                     "leadsmat_mu", "mitj_dies_mat_grau", "mitj_dies_mat_mu", "leads_org",
                     "leadsmat_org", "mitj_dies_mat_org", "leads_grau_org", "leads_mu_org",
                     "leadsmat_grau_org", "leadsmat_mu_org", "mitj_dies_mat_grau_org",
                     "mitj_dies_mat_mu_org", "IS", "IN", "N", "NM", "C", "T", "clicksIS",
```

```
                    "clicksIN", "clicksN", "clicksNM", "clicksC", "clicksT", "impIS",
                    "impIN", "impN", "impNM", "impC", "impT")

#El siguiente apartado es para graficar por pantalla.

ggfi <- ggplot(weekdata, mapping = aes(date, clicksNM)) +
  geom_line(aes(y = mat, color="matr cules")) +
  geom_line(aes(y = clicksNM, color="Clics_NM")) +
  geom_line(aes(y = sessio_org, color="Sessions_org"))

ggplotly(ggfi)
```

Analisi.R

```
library(corrgram)
library(corrplot)
library(ggplot2)
library(plotly)
library(stringr)
library(dplyr)
library(normalr)
library(rJava)
library(xlsx)
library(lubridate)
library(forecast)
library(tseries)
library(stats)
library(astsa)
library(fitdistrplus)
library(univariateML)


################MATRIZ CORRELACION################
##################################################

corr<- data.frame(weekdata$clics, weekdata$impressions,
    weekdata$clicksNM, weekdata$clicksT, weekdata$impNM,
    weekdata$leads, weekdata$leadsmat,
    weekdata$mat, weekdata$sessio_org)
cor <- corrgram(corr)
corrplot(cor, method = "number", type = "upper")

ggCT.week <- ggplot(weekdata, mapping = aes(date, clics, colour=Series)) +
  geom_line(aes(y = weekdata$clicksNM,
  color="Clics Navegacionales Marca")) +
  scale_y_continuous(breaks=0)+
  geom_line(aes(y=weekdata$clics, color="Clics"))+
  ggtitle("")+ ylab("Clics")+ xlab("Fecha")
ggplotly(ggCT.week)

ggCT.week <- ggplot(weekdata, mapping = aes(date, clics, colour=Series)) +
  geom_line(aes(y = weekdata$impNM,
  color="Impresiones Navegacionales Marca")) +
  scale_y_continuous(breaks=0)+
  geom_line(aes(y=weekdata$impressions, color="Impresiones"))+
  ggtitle("")+ ylab("Impresiones")+ xlab("Fecha")
ggplotly(ggCT.week)

ggCT.week <- ggplot(weekdata, mapping = aes(date, clics, colour=Series)) +
  geom_line(aes(y = weekdata$sessio_org, color="Sesiones")) +
  scale_y_continuous(breaks=0)+
  geom_line(aes(y=weekdata$clicksNM, color="Clics Navegacionales Marca"))+
  ggtitle("")+ ylab("")+ xlab("Fecha")
ggplotly(ggCT.week)

ggCT.week <- ggplot(weekdata, mapping = aes(date, clics, colour=Series)) +
  geom_line(aes(y = weekdata$clicksT, color="Clics transaccionales")) +
  scale_y_continuous(breaks=0)+ geom_line(aes(y=weekdata$leads, color="Leads"))+
  ggtitle("")+ ylab("")+ xlab("Fecha")
ggplotly(ggCT.week)

ggCT.week <- ggplot(weekdata, mapping = aes(date, clics, colour=Series)) +
  geom_line(aes(y = weekdata$clicksT, color="Clics transaccionales")) +
  scale_y_continuous(breaks=0)+
  geom_line(aes(y=weekdata$leadsmat, color="Leads con matr cula"))+
  ggtitle("")+ ylab("")+ xlab("Fecha")
ggplotly(ggCT.week)

ggCT.week <- ggplot(weekdata, mapping = aes(date, clics, colour=Series)) +
  geom_line(aes(y = weekdata$clicksT, color="Clics transaccionales")) +
```

```
  scale_y_continuous(breaks=0)+ geom_line(aes(y=weekdata$mat, color="Matr culas"))+
  ggtitle("")+ ylab("")+ xlab("Fecha")
ggplotly(ggCT.week)

#############Analisis Parte 1#################
#############################################

ggCT.day <- ggplot(daydata, mapping = aes(date, clicksT)) +
  geom_line()+scale_y_continuous(breaks=0)+ylab("Clics Transaccionales")+
  xlab("Fecha")+ggtitle("Serie Temporal con frecuencia diaria")
ggplotly(ggCT.day)

ggCT.week <- ggplot(weekdata, mapping = aes(date, clicksT)) +
  geom_line()+scale_y_continuous(breaks=0)+ylab("Clics Transaccionales")+
  xlab("Fecha")+ggtitle("Serie Temporal con frecuencia semanal")
ggplotly(ggCT.week)

freq <- c(1:115)
recta <- lm(weekdata$clicksT ~ freq)
summary(recta)
tendencia <- freq*1.7540+315.3899
CTsintendencia <- weekdata$clicksT-tendencia
tendencia0 <- tendencia-tendencia
semana <- c(46:52, 1:52, 1:52, 1:4)
analisi <- data.frame(weekdata$date, freq, semana, weekdata$clicksT,
                      tendencia, CTsintendencia, tendencia0)
names(analisi) <- c("data", "frecuencia","semana", "CT", "tendencia",
                    "CTsintendencia", "tendencia0")

ggCT.week <- ggplot(weekdata, mapping = aes(date, clicksT, colour=Rectas)) +
  geom_line(aes(color="Clics Transaccionales"))+scale_y_continuous(breaks=0)+
  geom_line(aes(y = analisi$tendencia, color="Tendencia"))+
  ylab("Clics Transaccionales")+ xlab("Fecha")+
  ggtitle("Serie Temporal con tendencia lineal")
ggplotly(ggCT.week)

ggCT.week <- ggplot(weekdata, mapping = aes(date, clicksT, colour=Rectas)) +
  scale_y_continuous(breaks=0)+
  geom_line(aes(y = analisi$tendencia0, color="Tendencia 0")) +
  geom_line(aes(y = analisi$CTsintendencia, color="Clics T sin tendencia"))+
  scale_y_continuous(breaks=0)+ggtitle("Serie Temporal con tendencia 0")+
  ylab("Clics Transaccionales")+ xlab("Fecha")
ggplotly(ggCT.week)

ciclo <- aggregate(analisi$CTsintendencia, by=list(analisi$semana), FUN=mean)

ggCT.week <- ggplot(ciclo, mapping = aes(Group.1, x)) +
  geom_line() +scale_y_continuous(breaks=0)+scale_x_continuous(breaks=0)+
  ggtitle("Ciclo de un a o")+ ylab("Clics Transaccionales")+ xlab("A o")
ggplotly(ggCT.week)

b=45
for (x in 1:115) {
  b=b+1
  print (b)
  print (x)
  analisi$cicl[x]<- ciclo$x[b]
  if (b==52) {b=0}
}
analisi$cicl <- as.numeric(analisi$cicl)

ggCT.week <- ggplot(analisi, mapping = aes(data, cicl)) +
  geom_line(aes(y = analisi$cicl)) +scale_y_continuous(breaks=0)+
  ggtitle("Componente Estacionaria")+ ylab("Clics Transaccionales")+
  xlab("Fecha")
ggplotly(ggCT.week)

analisi$aleatorio <- analisi$CTsintendencia-analisi$cicl
```

```
ggCT.week <- ggplot(analisi, mapping = aes(data, cicl, colour=Series)) +
  geom_line(aes(y = analisi$CTsintendencia, color="Observado sin tendencia")) +
  geom_line(aes(y = analisi$cicl, color="Componente Estacionaria"))+
  scale_y_continuous(breaks=0)+
  ggtitle("Comparativa")+ ylab("Clics Transaccionales")+ xlab("Fecha")
ggplotly(ggCT.week)

ggCT.week <- ggplot(analisi, mapping = aes(data, cicl, colour=Series)) +
  geom_line(aes(y = analisi$CTsintendencia, color="Observado sin tendencia")) +
  geom_line(aes(y = analisi$cicl, color="Componente Estacionaria"))+
  scale_y_continuous(breaks=0)+
  geom_line(aes(y=analisi$aleatorio, color="Componente Aleatoria"))+
  ggtitle("Comparativa")+ ylab("Clics Transaccionales")+ xlab("Fecha")
ggplotly(ggCT.week)


ggCT.week <- ggplot(analisi, mapping = aes(data, cicl)) +
  geom_line(aes(y=analisi$aleatorio))+scale_y_continuous(breaks=0)+
  ggtitle("Componente Aleatorio")+ ylab("Clics Transaccionales")+ xlab("Fecha")
ggplotly(ggCT.week)

ggCT.week <- ggplot(analisi, mapping = aes(data, aleatorio)) +
  geom_bar(stat = "identity")+scale_y_continuous(breaks=0)+
  ggtitle("Componente Aleatorio")+ ylab("Clics Transaccionales")+ xlab("Fecha")
ggplotly(ggCT.week)


############Analisis Parte 2################
############################################


ggCT.week <- ggplot(analisi, mapping = aes(data, cicl)) +
  geom_line(aes(y=analisi$aleatorio))+scale_y_continuous(breaks=0)+
  ggtitle("Componente Aleatorio")+ ylab("Clics Transaccionales")+ xlab("Fecha")
ggplotly(ggCT.week)

acfA <- acf(analisi$aleatorio)
pacfA <- pacf(analisi$aleatorio)
plot(acfA, main = "ACF del componente aleatorio")
plot(pacfA, main = "PACF del componente aleatorio")

Box.test(analisi$aleatorio, type = "Box-Pierce")
Box.test(analisi$aleatorio, type = "Ljung")
shapiro.test(analisi$aleatorio)
hist(analisi$aleatorio, main="Histograma del componente aleatorio", xlab="", breaks=30)

arma <- ts(analisi$aleatorio)
myarma <- auto.arima(arma, stepwise = T, approximation = F, trace= T)
myarma
ggCT.week <- ggplot(analisi, mapping = aes(data, cicl, colour=Series)) +
  geom_line(aes(y=arma, color="Componente aleatorio"))+
  geom_line(aes(y = myarma$fitted, color="Ajuste AR(1)"))+
  scale_y_continuous(breaks=0)+
  ggtitle("Comparativa tras ajustar un AR(1)")+
  ylab("Clics Transaccionales")+ xlab("Fecha")
ggplotly(ggCT.week)

myarmaf <- forecast(myarma, h = 30)
autoplot(myarmaf) + geom_forecast(h=30)

analisi$residuos <- myarma$residuals

ggCT.week <- ggplot(analisi, mapping = aes(data, residuos, colour=Series)) +
  geom_line(aes(y=residuos, color="Residuos"))+
  scale_y_continuous(breaks=0)+
  ggtitle("Residuos del ajuste AR(1)")+
  ylab("Clics Transaccionales")+ xlab("Fecha")
ggplotly(ggCT.week)

acfR <- acf(myarma$residuals)
```

```
pacfR <- pacf(myarma$residuals)
plot(acfR, main = "ACF de los residuos")
plot(pacfR, main = "PACF de los residuos")

summary(myarma$residuals)
var(myarma$residuals)
sd(myarma$residuals)

Box.test(myarma$residuals, type = "Ljung")
Box.test(myarma$residuals, type = "Box-Pierce")
shapiro.test(myarma$residuals)# p < 0.05, distribuci   n no normal.
hist(myarma$residuals, main="Histograma de los residuos", xlab="Residuos", breaks=30)

# acf(myarma$residuals^2)
# pacf(myarma$residuals^2)

#estudio distribuci n residuos
residuos <- as.numeric(myarma$residuals)
pos_res <- residuos+155
comparacion_aic <- AIC(
  mlbetapr(pos_res),
  mlexp(pos_res),
  mlinvgamma(pos_res),
  mlgamma(pos_res),
  mllnorm(pos_res),
  mlrayleigh(pos_res),
  mlinvgauss(pos_res),
  mlweibull(pos_res),
  mlinvweibull(pos_res),
  mllgamma(pos_res))

distribucion <- fitdist(pos_res, distr = "weibull")
summary(distribucion)
plot(distribucion)

#prediccion

myarma <- auto.arima(arma, stepwise = T, approximation = F, trace= T)
myarmaf <- forecast(myarma, h = 30)
autoplot(myarmaf) + geom_forecast(h=30)

freq2 <- c(1:145)

tendencia2 <- freq2*1.7540+315.3899

semana2 <- c(46:52, 1:52, 1:52, 1:34)
analisi2 <- data.frame(freq2, semana2, tendencia2)
names(analisi2) <- c("frecuencia","semana", "tendencia")

b=45
for (x in 1:145) {
  b=b+1
  print (b)
  print (x)
  analisi2$estacional[x]<- ciclo$x[b]
  if (b==52) {b=0}
}
analisi2$estacional <- as.numeric(analisi2$estacional)


mean <- as.numeric(myarmaf$mean)
upper <- as.numeric(myarmaf$upper)
hi <- upper[c(1:30)]
hi2 <-upper[c(31:60)]
lower <- as.numeric(myarmaf$lower)
low <- lower[c(1:30)]
low2 <-lower[c(31:60)]
alea <- as.numeric(analisi$aleatorio)
analisi2$aleatorio <- c(alea, mean)
analisi2$hi <- c(alea, hi)
```

```
analisi2$low <- c(alea, low)
analisi2$hi2 <- c(alea, hi2)
analisi2$low2 <- c(alea, low2)

analisi2$prediccion <- analisi2$tendencia+analisi2$estacional+analisi2$aleatorio
analisi2$predhi <- analisi2$tendencia+analisi2$estacional+analisi2$hi
analisi2$predlow <- analisi2$tendencia+analisi2$estacional+analisi2$low
analisi2$predhi2 <- analisi2$tendencia+analisi2$estacional+analisi2$hi2
analisi2$predlow2 <- analisi2$tendencia+analisi2$estacional+analisi2$low2


ggCT.week <- ggplot(analisi2, mapping = aes(frecuencia, prediccion, colour=Series)) +
  geom_line(aes(y = analisi2$predhi, color="80%"))+
  geom_line(aes(y=analisi2$predlow, color="80%"))+
  geom_line(aes(y = analisi2$predhi2, color="95%"))+
  geom_line(aes(y=analisi2$predlow2, color="95%"))+
  scale_y_continuous(breaks=0)+
  geom_line(aes(y = analisi2$prediccion, color="predicci n")) +
  ggtitle("Predicci n")+ ylab("Clics_Transaccionales")+ xlab("Fecha")
ggplotly(ggCT.week)


tsclickst <- ts(as.numeric(weekdata$clicksT), frequency = 52, start = c(2017,46))
myarima <- auto.arima(tsclickst, stepwise = T, approximation = F, trace= T)
matarima <- forecast(myarima, h = 30)
autoplot(matarima) + geom_forecast(h=30)
```

Referencias

[1] TUKEY J.W., The Future of Data Analysis, The Annals of Mathematical Statistics, Vol.33, No.1 (March 1962).

[2] DOUGLAS L., 3D Data Management: Controlling Data Volume, Velocity and Variety. Gartner. Consultado el 6 de febrero de 2001.

[3] VILA M.A., SANCHEZ D., ESCOBAR L, Relaciones Causales en Reglas de Asociación, XII Congreso Español sobre tecnologías y lógica Fuzzy, 2004.

[4] TANNER K., Google marketing platform products: 20 questions, 31 de enero de 2020. https://www.inmarketingwetrust.com.au/google-marketing-platform-products-questions/

[5] JOSHUA HARDWICK, Intención de búsqueda: el "factor de posicionamiento" que se pasa por alto y por el que deberías estar optimizando en 2019, 2019. https://ahrefs.com/blog/es/intencion-de-busqueda/

[6] BALL P, Counting Google searches predicts market movements, Nature, 26 de abril de 2013.

[7] https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html

[8] PEÑA D., Análisis de series temporales, Alianza Editorial. Madrid, 2010.

[9] BROCKWELL P.J AND DAVIS R.A, Time Series: Theory and Methods, Colorado State University, 1987.

[10] PAUL S. P. COWPERTWAIT AND ANDREW V. METCALFE, Introductory time series with R, Springer, 2009.

[11] BOX G.E.P AND JENKINS G.M, Time Series Analysis: Forecasting and Control, Revised Edition Holden-Day, San Francisco 1976.

[12] BROCKWELL P.J AND DAVIS R.A, Introduction to Time Series and Forecasting, Springer, Third Edition, 2016.

[13] ROB J HYNDMAN AND GEORGE ATHANASOPOULOS, Forecasting: Principles and Practice, 2 edition (May 6, 2018) https://otexts.com/fpp3/

[14] SHUMWAY, R.H AND STOFFER D.S, Time series analysis and its applications with R examples, Second Edition. Springer, 2006.

[15] BOX G.E.P., PIERCE D.A., Distribution of residual correlations in autoregressive-integrated moving average time series models, Journal of the American Statistical Association, 1970.

[16] AKAIKE H., .ºn entropy maximization principle", Applications of Statistics, North-Holland, Amsterdam, 1977, pp. 27–41.

[17] PAPOULIS P., Probability, Random Variables, and Stochastic Processes, McGraw-Hil, 2002.

[18] HYNDMAN, R.J, Forecasting functions for time series and linear models. ,R package version 8.0 http://github.com/robjhyndman/forecast.